ENIGMA · ACADEMIC PUBLISHING

Mathematics & AI

Volume 1 · Number 2

2026 · MathAI 2026 Selected Papers

Published by IAIC AI RESEARCH & TRADING — FZCO, Dubai, United Arab Emirates

enigma.ist/j/mathematics-ai

Every article in this issue is free to read, free to download and free to publish.

CONTENTS

Article 1004

AI-Based Detection of Unwanted Behavior: The Paradoxical Effect of Standard Data Augmentation in Video Surveillance

Emelianov, M.V.

DOI 10.66693/mathai.1004

Article 1005

A MONOTONE SYSTEM GENERATOR FOR SOLVING BIG DATA AGGREGATION PROBLEMS

Adilova, F.; Davronov, R.

DOI 10.66693/mathai.1005

Article 1006

Deep Learning for Educational Video Analysis: Benchmarking ASR Systems and Pipeline Optimization

Zuev, G.; Kantonistova, E.

DOI 10.66693/mathai.1006

Article 1007

The Evolution of Mind: Emergence of Collective Intelligence through Logical-Probabilistic Knowledge Dynamics in Multi-Agent ENIGMA Metaverse Ecosystems

Nechesov, A.; Barykin, S.; Ruponen, J.

DOI 10.66693/mathai.1007

Article 1009

OLORA+: A HYBRID APPROACH TO PARAMETER- EFFICIENT FINE-TUNING OF LARGE LANGUAGE MODELS

S.I.Kushmuratov, F.T. Adilova, R.R. Davronov

DOI 10.66693/mathai.1009

Article 1013

Interpretable Emotion Attribution in Social Graphs: A Comparative Analysis of Rule-Based, Transformer, and LLM Models

Gidado, U.; Kolonin, A.

DOI 10.66693/mathai.1013

Article 1014

Hybrid Bi-Level Index for Shortest Paths in Temporal Networks

Vasilev, M.

DOI 10.66693/mathai.1014

Article 1015

A Systematic Study of Gate Functions in Soft Adaptive Policy Optimization

Denisov, E.; Glazyrina, S.; Kryzhanovskiy, M.; Ischenko R.

DOI 10.66693/mathai.1015

Article 1017

Improving Fairness in AI-Powered Recrutiment: An Interpretable Resume Screening System

Agapova, N. A.; Lukmanov, R. A.

DOI 10.66693/mathai.1017

Article 1020

Explainable AI for Mathematics: Proofs as Code with Knowledge Graph and Domain Ontology Support

Khalov A.; Ataeva, O.; Tuchkova, N.

DOI 10.66693/mathai.1020

Article 1022

Hybrid UAV Hazard Detection Approach Based on Open-Vocabulary Detection and MIVAR Expert System

Lamcev, O.; Chernyadiev, I.; Maksimova, A.

DOI 10.66693/mathai.1022

Article 1023

Implementation of a Cryptographic Hash Function Based on a Deep Neural Network.

Iatsenko, D; Gorin, D

DOI 10.66693/mathai.1023

Article 1024

Mathematics of natural Intelligence

Evgenii Vityaev

DOI 10.66693/mathai.1024

Article 1025

Detecting Hallucinations In LLM Responses Using Token-level Log-probability Signals

Eliseev, V.; Maksimova, A.

DOI 10.66693/mathai.1025

Article 1026

Estimating Importance of Highly Correlated Features Using Matrix Factorization

Minkin, A.

DOI 10.66693/mathai.1026

Article 1030

Operator Learning for High-Dimensional Symplastic Growth Dynamics with Stochastic Cell Division

Ulyana Zubairova

DOI 10.66693/mathai.1030

Article 1032

Dynamic Data Classification Based on a Semi-Supervised Local Poisson Label Propagation Method

Constantin

DOI 10.66693/mathai.1032

Article 1036

Aligning the Number of Parameters with the Number of Linear Regions for Improved Neural Network Approximation

Podoprosvetov, A.; Smolin, V.; Sokolov, S.

DOI 10.66693/mathai.1036

Article 1038

Task-Based Engineering

Sviridenko, D. I.

DOI 10.66693/mathai.1038

This file is a print-ready copy of the complete issue, assembled from the published articles. It is provided free of charge and may be printed and distributed without permission.

Generated 2026-11-03. The authoritative version of every article is the one at its registered DOI.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1004 DOI: 10.66693/mathai.1004
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

AI-Based Detection of Unwanted Behavior: The Paradoxical Effect of Standard Data Augmentation in Video Surveillance

Maksim V. Emelianov
Department of Mechanics and Mathematics
Novosibirsk State University (NSU)
Novosibirsk, Russia
m.emelyanov1@g.nsu.ru

Abstract

Public spaces and commercial environments face persistent challenges regarding human misconduct. Traditional surveillance remains passive, while manual monitoring is labor-intensive and inefficient. Consequently, automating the detection of unwanted behavior via digital assistants is essential. This study explores the application of deep learning models to identify such behavior and analyzes the impact of data augmentation on model performance.

We utilized the UCF-Crime dataset (1,900 videos, 13 classes) and constructed a novel custom dataset comprising 5,236 videos with a focus on "violent behavior." Algorithms based on ResNet3D and DenseNet-RNN were trained on both original and augmented versions of these datasets. On the UCF-Crime dataset, the DenseNet-RNN model showed stability only in detecting the "arson" class. However, the model's stability and recognition quality improved significantly when trained on the custom dataset.

Crucially, our experiments demonstrated that data augmentation negatively impacted the recognition quality on the custom dataset ($F(1, 36) = 7.40, p = 0.01$). Specifically, the ResNet3D method exhibited a significant degradation in performance, with the AUC dropping from 0.83 (95% CI: $[0.80 - 0.86]$) before augmentation to 0.73 (95% CI: $[0.69 - 0.77]$) post-augmentation. The DenseNet-RNN method showed a similar downward trend (AUC 0.82, 95% CI: $[0.79 - 0.85]$ vs. AUC 0.75, 95% CI: $[0.71 - 0.79]$). These findings suggest that the blind application of standard augmentation procedures—a common industry practice for both static images and video frames—can lead to a paradoxical decline in detection quality. This highlights the urgent need to critically reevaluate how augmentations are selected and to develop task-specific methodologies that ensure transformations actually improve, rather than hinder, model efficacy in behavioral recognition.

1 Introduction

In modern society, public spaces and commercial environments frequently face challenges related to visitor misconduct. These unwanted behaviors range from minor rule violations to severe criminal activities, such as vandalism, assault, and physical violence [1]. Such incidents inevitably lead to property damage and significant safety risks, making the automated monitoring of mass-gathering areas a critical priority for public security.

To mitigate these risks, security stakeholders heavily rely on Closed-Circuit Television (CCTV) systems to monitor human activity. Despite their widespread deployment, traditional surveillance systems remain inherently passive. Due to the limitations of human attention and the cognitive fatigue associated with monitoring numerous screens simultaneously, manual surveillance is increasingly labor-intensive and economically inefficient. Consequently, CCTV is often relegated to a forensic tool used only after an incident has occurred, thereby diminishing the return on costly security investments.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1004 DOI: 10.66693/mathai.1004
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

To enhance operational efficiency, there is a critical need for intelligent video surveillance systems capable of autonomously detecting unwanted behaviors in real-time. Recent advancements in deep learning, particularly Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have shown significant promise in video processing and action recognition tasks. However, predicting human behavior in complex environments remains a formidable challenge. The performance of these intelligent systems heavily relies on the quality of the training data and the chosen network architectures. Existing datasets often suffer from low resolution, extreme class imbalance, and a lack of precise spatial-temporal annotations, which collectively impede the generalization capabilities of predictive models.

In this study, we propose a neural network-based framework for the detection of unwanted human behavior in general surveillance settings. Our contributions are threefold:

- We address the limitations of existing public datasets (such as UCF-Crime) by constructing and distilling a novel custom dataset of 5,236 video clips, heavily annotated and focused specifically on violent behavior.
- We adapt and evaluate two distinct architectures—a hybrid DenseNet-RNN model and a ResNet3D model—for the task of behavioral screening in constrained real-world computing environments.
- We systematically investigate the impact of online data augmentation on model performance. Surprisingly, our results reveal a paradoxical effect: standard augmentation techniques significantly degrade the detection capabilities of the models in this specific domain.

The remainder of this paper is structured as follows: Section 2 reviews existing literature and datasets. Section 3 details our proposed methodology, including dataset preparation and network architectures. Section 4 presents our experimental results, followed by a discussion of the augmentation paradox in Section 5. Finally, Section 6 concludes the paper.

2 Related Work

Video Anomaly Detection (VAD) and Datasets. The detection of anomalous or unwanted behavior in video streams has been extensively studied [7]. The efficacy of predictive models heavily depends on the diversity and quality of the training datasets. Several public datasets have been introduced to address this, including the Avenue Dataset [6] and the DCSASS dataset. Notably, the UCF-Crime dataset [8] provides a large-scale collection of 1,900 untrimmed surveillance videos covering 13 anomaly classes (e.g., arson, assault, shoplifting). While comprehensive, UCF-Crime and similar datasets often contain excessive background noise, lack precise temporal boundary annotations, and suffer from severe class imbalances, making fine-grained behavior detection challenging.

Action Recognition Architectures. Traditional approaches to video analysis have transitioned from hand-crafted features to deep learning architectures. Standard 2D Convolutional Neural Networks (CNNs) excel at spatial feature extraction but fail to capture temporal dynamics. To address this, hybrid models combining CNNs with Recurrent Neural Networks (e.g., LSTMs) have become prominent for sequential data modeling. More recently, 3D Convolutional Networks (like ResNet3D) have demonstrated superior performance by simultaneously learning spatial and temporal representations directly from video volumes. While recent state-of-the-art methods (such as Video Transformers) achieve exceptional accuracy on public benchmarks, they often incur prohibitive computational costs. In contrast, traditional architectures like DenseNet [4] and ResNet [3] remain highly effective and are significantly less computationally demanding, which is an absolute necessity for deployment on resource-constrained edge devices in real-world surveillance systems.

3 Methodology

Our approach to detecting unwanted behavior involves evaluating two computationally efficient deep learning architectures under various data conditions. We specifically focus on the impact of data augmentation and the refinement of training datasets.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1004 DOI: 10.66693/mathai.1004
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

3.1 Datasets and Data Distillation

Initially, we utilized the publicly available **UCF-Crime Dataset**, which encompasses 128 hours of footage across 13 classes of anomalous behavior. However, preliminary experiments revealed that the presence of prolonged normal segments within these videos diluted the learning signal. The lack of explicit spatial-temporal localization for the individuals exhibiting the unwanted behavior hindered the network's ability to extract relevant features.

To overcome these limitations, we constructed a **Custom Distilled Dataset**. This dataset specifically targets a single, well-defined class of unwanted behavior: "violent behavior" (encompassing fights and physical aggression). It consists of 5,236 curated video clips ranging from 5 to 60 seconds. The data underwent a rigorous "distillation" process, incorporating precise temporal trimming to isolate the anomaly and spatial tracking via YOLO to focus the region of interest (ROI) on the acting individuals. Importantly, YOLO proved highly robust to motion blur during fast physical strikes, retaining human silhouettes effectively. Natural occurrences of individuals briefly leaving the frame or being occluded were preserved in the dataset, ensuring that the tracking process did not introduce artificial selection bias (e.g., dropping hard-to-detect violent frames). This dataset provides a highly balanced, noise-reduced environment for training.

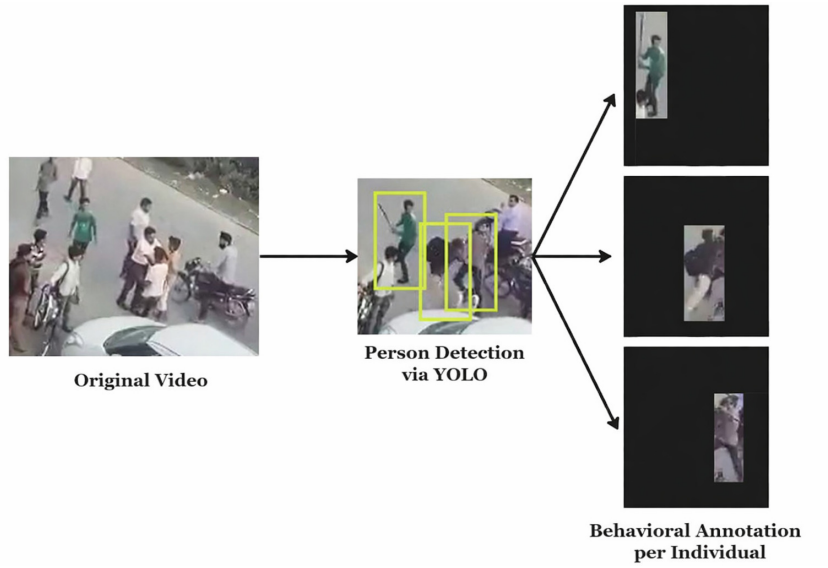


Figure 1: Spatial-temporal annotation and distillation process of the custom dataset, highlighting the region of interest (ROI) and temporal boundaries of violent behavior.

3.2 Network Architectures

We deployed two distinct architectures optimized for video sequence analysis:

1. DenseNet-RNN Hybrid Model: This architecture processes videos sequentially. Each video is divided into frames, and a pre-trained 121-layer DenseNet operates as a spatial feature extractor, benefiting from dense connectivity to maximize feature reuse. The resulting 1024-dimensional feature vectors from sequences of four consecutive frames are fed into a 2-layer Long Short-Term Memory (LSTM) network [2] (hidden state dimension of 512). The LSTM models the temporal evolution of the features, outputting a context vector that is passed through a fully connected layer for final classification.

2. ResNet3D: To inherently capture spatiotemporal features, we utilized a modified ResNet-2p1d architecture with a depth of 152 layers. Unlike the 2D convolutions in standard ResNets, the 3D convolutional kernels operate across spatial dimensions and the temporal time-step. We optimized the input window to process sequences of four frames, leveraging residual connections to facilitate deep feature extraction without the vanishing gradient problem.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1004 DOI: 10.66693/mathai.1004
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

3.3 Training Strategy and Online Augmentation

Both models were trained using a standard split strategy with Cross-Entropy Loss (or Binary Cross-Entropy with Logits Loss for the custom dataset) and the Adam optimizer [5]. To prevent overfitting, we applied L_2 regularization.

A critical aspect of our methodology was the application of **Online Data Augmentation**. To artificially expand the training distribution, we applied a stochastic pipeline during the training phase. Transformations included:

- Severe spatial transformations (180° rotation) and subtle geometric shifts (random tilt/rotation, horizontal flipping).
- Random scaling ($0.8 - 1.2$) and cropping ($100 - 200$ pixels).
- Photometric distortions including additive random noise, brightness adjustments, and grayscale conversion.

This specific bundle of transformations was deliberately chosen because it represents a standard, widely adopted pipeline in modern image recognition workflows, where it is often applied blindly to increase data volume. While these techniques are ubiquitous in computer vision to improve generalization, our subsequent experiments critically evaluate their efficacy in the specific context of automated behavioral screening.

4 Experiments and Results

To rigorously evaluate the proposed architectures and understand the impact of data augmentation, we conducted a series of cross-validation experiments. Model performance was primarily assessed using the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) to provide a threshold-invariant metric. We report the AUC alongside its 95% Confidence Interval (CI). Additionally, we captured momentary metrics—Accuracy, Precision, Recall, F1-Score, and Specificity—to provide a comprehensive performance profile.

4.1 Baseline Evaluation on the UCF-Crime Dataset

Initial experiments were conducted using the multi-class, unaugmented UCF-Crime dataset to establish a baseline for the DenseNet-RNN hybrid model. The goal was to classify 13 distinct types of anomalous behavior alongside normal surveillance footage.

The results indicated severe instability and generally poor predictive performance across most classes, yielding an average AUC of approximately 0.51. The model struggled significantly with minority classes; for instance, "Assault" and "Vandalism" demonstrated highly unstable AUC trajectories during cross-validation, occasionally dropping below 0.50, effectively indicating inverted or random predictions. Notably, out of the 13 categories, the model exhibited stability only in detecting the "Arson" class (see Figure 2). We hypothesize that the distinct, persistent visual signature of fire allowed the spatial feature extractor to consistently identify this anomaly, despite the background noise.

Overall, this exploratory evaluation on UCF-Crime confirmed that without explicit spatial-temporal bounding, traditional architectures struggle to learn from long, untrimmed videos dominated by background frames. Rather than serving as a direct benchmark comparison, this initial failure necessitated the shift to our Custom Distilled Dataset, changing the task to a focused, binary behavior classification problem.

4.2 Performance on the Custom Distilled Dataset

By transitioning to the Custom Distilled Dataset—which isolated the "violent behavior" class using precise spatial and temporal tracking—we observed a marked improvement in model stability and absolute performance metrics, independent of the applied architecture (as illustrated by the tight clustering of black squares in Figure 2).

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1004 DOI: 10.66693/mathai.1004
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

When trained without data augmentation, the **ResNet3D** architecture achieved robust classification results. The model recorded an AUC of 0.83 (95% CI: $[0.80 - 0.86]$). The momentary metrics further validated its efficacy as a screening tool, achieving an Accuracy of 80.3%, Precision of 65.0%, Recall of 65.8%, and an F1-Score of 65.4%.

Similarly, the **DenseNet-RNN** model without augmentation demonstrated comparable baseline efficiency, reaching an AUC of 0.82 (95% CI: $[0.79 - 0.85]$). This indicates that both spatial-temporal sequence modeling and 3D convolutions are highly viable approaches for this domain when trained on refined data.

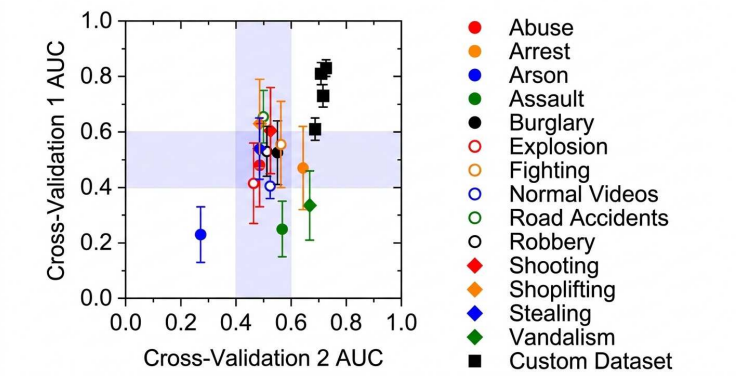


Figure 2: Cross-validation AUC stability comparison. Scatter plot shows the relationship between two cross-validation iterations. Classes from the UCF-Crime dataset display wide dispersion (low stability), whereas the models trained on the Custom Distilled Dataset (represented by black squares) exhibit tight clustering in the upper-right quadrant, indicating both high accuracy and robust stability regardless of the specific architecture.

4.3 The Augmentation Paradox: Statistical Analysis

In standard ML workflows, data augmentation is universally applied under the assumption that artificially expanding the training distribution inherently improves model generalization. To test this assumption in the context of behavioral screening, we subjected the Custom Distilled Dataset to the aggressive stochastic augmentation pipeline detailed in Section 3.

Paradoxically, the application of this standard bundle resulted in a statistically significant degradation of detection quality. A repeated-measures analysis of variance revealed a strong negative main effect of augmentation on the models' predictive capabilities ($F(1, 36) = 7.40, p = 0.01$).

Table 1: Impact of Data Augmentation on Model Performance (Custom Dataset)

Architecture	Condition	AUC	95% CI	Accuracy
ResNet3D	No Augmentation	**0.83**	$[0.80 - 0.86]$	**0.803**
ResNet3D	With Online Augmentation	0.73	$[0.69 - 0.77]$	0.666
DenseNet-RNN	No Augmentation	0.82	$[0.79 - 0.85]$	0.762
DenseNet-RNN	With Online Augmentation	0.75	$[0.71 - 0.79]$	0.697

As summarized in Table 1, the **ResNet3D** method experienced a steep decline in performance. The AUC dropped from 0.83 down to 0.73, and the confidence interval shifted completely downwards ($[0.69 - 0.77]$ post-augmentation). This performance degradation was statistically significant ($p = 0.02$). Furthermore, the momentary metrics for ResNet3D collapsed under augmentation: Accuracy fell to 66.6%, and Precision dropped severely to 49.7%, generating an unacceptable rate of false positives for a surveillance system.

The **DenseNet-RNN** architecture exhibited a parallel downward trend. The baseline AUC of 0.82 degraded to 0.75 under augmented conditions. While the isolated statistical drop for DenseNet-RNN

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1004 DOI: 10.66693/mathai.1004
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

was slightly less pronounced ($p = 0.17$), the consistent trajectory across entirely different network paradigms confirms the overarching negative impact.

These findings robustly demonstrate that standard spatial and pixel-level augmentations—rather than aiding the neural networks—corrupted the delicate spatial-temporal signatures required to identify human violence.

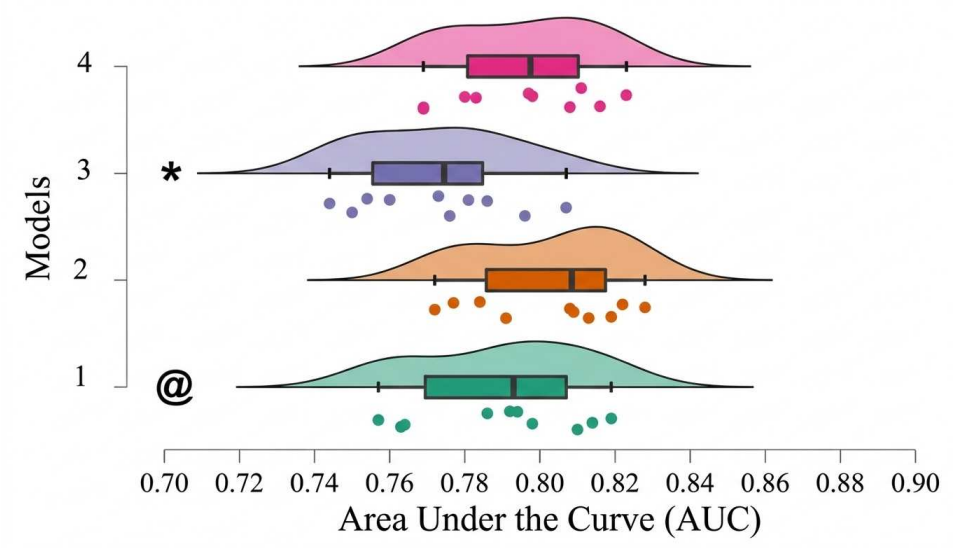


Figure 3: Density and boxplot distributions of AUC scores across different modeling approaches on the Custom Distilled Dataset. Models without data augmentation (Model 4: ResNet3D, Model 2: DenseNet-RNN) consistently outperform their counterparts trained with online data augmentation (Model 3: ResNet3D, Model 1: DenseNet-RNN). The application of augmentation causes a visible leftward shift, demonstrating the augmentation paradox. Symbols denote statistical comparisons against the respective unaugmented baselines: $*$ indicates $p = 0.02$ (Model 3 vs. Model 4), and @ indicates $p = 0.17$ (Model 1 vs. Model 2). The shaded curves represent the probability density of AUC values, while the embedded boxplots display the median, interquartile range (25th and 75th percentiles), and non-outlier boundaries.

5 Discussion

The objective of this study was to evaluate lightweight, robust architectures (DenseNet-RNN and ResNet3D) for the automated detection of unwanted behavior in general surveillance environments. Initially, we hypothesized that the combination of spatial and recurrent layers would yield high AUC scores (>0.80) on established public datasets like UCF-Crime. However, empirical results contradicted this expectation, yielding an average AUC of 0.51 across 13 classes. Detailed class-wise analysis revealed that only the "Arson" class was reliably detected. We attribute this exception to the persistent, high-contrast, and relatively static visual signature of fire, which the spatial feature extractor could easily isolate. Conversely, behaviors defined by complex, subtle human interactions (e.g., "Assault" or "Shoplifting") were lost in the unannotated, lengthy background footage of the UCF-Crime dataset.

This limitation justified the creation of our Custom Distilled Dataset, which focused exclusively on violent behavior with strict temporal and spatial bounding. As expected, baseline models trained on this distilled dataset demonstrated a substantial leap in both absolute performance and cross-validation stability.

Explaining the Augmentation Paradox. The most significant finding of this study is the paradoxical degradation of model performance upon the application of standard online data augmentation. While techniques such as geometric transformations and color-space shifts are universally

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1004 DOI: 10.66693/mathai.1004
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

recommended to prevent overfitting in static object recognition, they actively corrupted the learning process in our behavioral screening task.

We propose several mechanistic explanations for this phenomenon:

- **Disruption of Biomechanical Semantics:** Human violence involves specific physical postures and gravity-dependent dynamics (e.g., falling, striking). Severe geometric transformations, such as 180° rotations or extreme horizontal flipping, distort the natural physical constraints of these actions. The network is forced to learn physiologically impossible movements, thereby diluting the discriminative features of genuine violence.
- **Loss of High-Frequency Interaction Details:** Additive noise and low-resolution scaling obscure critical, high-frequency details—such as the rapid intersection of limbs or facial expressions—that differentiate an aggressive physical altercation from an innocuous interaction (e.g., a hug or a fast-paced game).
- **Color and Texture Distortion:** Grayscale conversion and aggressive brightness shifting destroy subtle physiological and environmental cues. For instance, the contrast between the clothing of two distinct individuals is a crucial feature for the spatial extractor to separate overlapping human bounds during a fight. Removing color information forces the network to rely solely on edges, which are often noisy in CCTV footage.

While it may seem intuitive in hindsight that severe geometric transformations (like 180° rotations) destroy biomechanical semantics, such transformations are routinely included in generic augmentation libraries and applied blindly across the industry to artificially boost dataset sizes. The paradox lies in the fact that a practice universally assumed to improve generalization actively harms it in this context. These findings emphasize that the blind application of standard augmentation bundles is fundamentally flawed for spatiotemporal data. We acknowledge that grouping highly destructive transformations with subtle photometric shifts into a single stochastic pipeline limits our ability to isolate the precise cause of the degradation. Conducting a detailed ablation study to isolate the independent effects of each transformation (e.g., random cropping vs. rotation) is the immediate next step in our future work. Ultimately, it is imperative to move away from arbitrary augmentation pipelines and establish rigorous, task-specific methodologies that verifiably enhance model efficiency in behavioral recognition.

6 Conclusion

In this paper, we addressed the challenge of detecting unwanted human behavior in video surveillance under constrained computational resources. Our contributions and findings are summarized as follows:

1. We demonstrated that public, untrimmed datasets (like UCF-Crime) are insufficient for fine-grained behavioral detection using standard sequence models, as they only reliably expose anomalies with distinct static signatures (e.g., arson).
2. To solve this, we curated and distilled a novel custom dataset comprising 5,236 annotated video clips focused on violent behavior, which drastically improved the baseline stability of both ResNet3D and DenseNet-RNN architectures.
3. We uncovered a significant *augmentation paradox*: the application of standard online data augmentation statistically degraded model performance ($p < 0.05$), lowering the AUC and destroying precision.

These results emphasize that compact neural network architectures are highly capable of screening unwanted behavior, provided the training data is carefully distilled. Future research should focus on developing domain-aware augmentation strategies specifically tailored for human kinematics and interaction dynamics, avoiding the destructive pitfalls of standard image-level transformations. Ultimately, these insights lay the groundwork for deploying modular, highly accurate AI assistants in general security systems, advocating for a paradigm shift from blind data augmentation to methodologically justified, task-specific data transformations.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1004 DOI: 10.66693/mathai.1004
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

REFERENCES

[1] M. A. Ansari and D. K. Singh. An expert video surveillance system to identify and mitigate shoplifting in megastores. *Multimedia Tools and Applications*, 81(16), 2022. https://doi.org/10.1007/s11042-021-11438-2.

[2] A. Graves. Long short-term memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012. https://doi.org/10.1007/978-3-642-24797-2_4.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. https://doi.org/10.1109/CVPR.2016.90.

[4] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. https://doi.org/10.1109/CVPR.2017.243.

[5] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Published at ICLR 2015.

[6] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 FPS in MATLAB. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2013. https://doi.org/10.1109/ICCV.2013.338.

[7] P. Sernani, N. Falcionelli, S. Tomassini, P. Contardo, and A. F. Dragoni. Deep learning for automatic violence detection: Tests on the AIRTLab dataset. *IEEE Access*, 9, 2021. https://doi.org/10.1109/ACCESS.2021.3131315.

[8] W. Sultani, C. Chen, and M. Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. https://doi.org/10.1109/CVPR.2018.00678.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1005 DOI: 10.66693/mathai.1005
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

A Monotone System Generator for Solving Big Data Aggregation Problems

F.T. Adilova[1]* and R.R. Davronov[1]

[1] *Laboratory of Biomedical Informatics, V.I. Romanovskiy Institute of Mathematics, Uzbekistan Academy of Sciences, 9 University Street, Tashkent 100174, Uzbekistan*

fatadilova@mathinst.uz, rifqat@gmail.com

Abstract

It is well known that the theory of monotone systems transforms clustering from a global optimization problem (which is often NP-hard) into a successive elimination problem solvable in polynomial time. The proposed approach requires minimal a priori information: specifying only the relationship measure of one object with a subset of objects. The algorithm guarantees an exact solution to the stated extremal problem. The approach is based on the concept of a monotone system $< A, F >$, where A is a finite set of objects, and $F(X)$ is an importance (or weight) function defined on subsets $(X \subseteq A)$. The monotonicity condition is: $F(X \setminus \{a\}) > F(X)$. We consider a generator of monotone systems. The proposed procedure for generating a family of monotone systems consists of two stages: i) constructing a set of transformation operators for a monotone system of a sufficiently general form, defined on the same initial set W, $|W| = N$; ii) constructing a set of basic functions on the set W. The desired generator is considered as a structure that generates compositional chains of operators over monotone systems selected as basis ones. The proposed extension of the class of basic functions for monotone systems is implemented in a class of Estimate Calculation Algorithms (ECA). A problem statement is formulated for defining a set of three types of basic functions in a monotone system in a class of estimator algorithms. Changing the sets of operators in the basic systems generates a family of monotone systems, which has a wide range of applications, for example, in genetic network analysis, natural language processing (NLP), image processing, and, in general, as a new tool for solving complex problems of structuring large data sets.

1 Introduction

The problem of cluster analysis lacks a clearly formalized criterion—all formalizations are subjective and not tied to model constructions. Therefore, fundamental decisions are called into question: the choice of features and their measurement scales, the selection of a measure of similarity between objects, and the interpretation of results. Generally, the clustering problem is formulated as a combinatorial extremization problem. In such problems, there are no effective procedures that would deliver a global extremum to the corresponding functional. Therefore, one can only hope to achieve a local extremum. These specificities are very strict restrictions for applied scientists who use broader concepts in their qualitative descriptions.

To overcome the limitations of known methods, increasingly complex combined analysis processes or specialized methods focused on specific types of information are being used to find precise solutions to the problem. Interesting results in solving this problem were once obtained using the theory of monotone systems, which represents a universal mathematical framework that allows for the removal of these restrictions at the level of problem formulation in general.

Here, clustering is viewed not as a partition of space, but as a process of sequentially identifying "cores" based on certain connectivity functions. It is based on the concept of a monotone system

*Corresponding author.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1005 DOI: 10.66693/mathai.1005
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

$\langle A, F\rangle$, where A is a finite set of objects, and $F(X)$ is the significance (or weight) function defined on subsets ($X \subseteq A$). The monotonicity condition is: $F(X \setminus \{a\}) > F(X)$. The motivation for this study was the desire to develop a method that is sufficiently rich in terms of covering the required number of objects under study, while at the same time reducing the complexity of finding the function. We see the solution as the construction of an ordered sequence of subclasses, in which each subsequent subclass contains functions with a more stringent assessment of the similarity of an element to a subset. A well-known method of this type is the method of Estimate Calculation Algorithms (ECA), which belongs to the school of Academician Yu. I. Zhuravlev (Zhuravlev et al., 1974).

2 Relevant Works

The increasingly recognized problem of the current stage of cluster analysis development has generated many accessible programs implementing different methods that generate different cluster structures using the same data. Which structure should be adopted as the final solution? In the context of discrete data analysis (the school of I. Muchnik and J. Mullat), monotone systems provide a unique clustering tool that is fundamentally different from classical methods like k-means.

To solve clustering problems, Mullat (1976) proposed an approach called a monotone cluster. The approach is based on assessing the relationships between objects and sets of objects (Kuznetsov and Muchnik, 1982; Aaremaa, 1986). The idea behind this approach is to use a monotonic similarity function between all objects and subsets as input. In a typical situation, such a function can easily be determined based on data of any nature, including digitized text or images.

The functions $F(X)$ can be maximized in a "greedy" manner, selecting the best candidate at each step, which leads to an "optimal" sequence of objects that determines not only the "core"—the set S that maximizes $F(S)$ as a fragment of this sequence—but also the set of its "shells"—the enclosing fragments. The weakest-link functions are only those that satisfy the so-called quasiconvexity condition for all $S, T \subset I$ associated with order structures such as finite "convex geometries" (Mirkin and Muchnik, 2002).

The basic idea is to define a connectivity function that evaluates the "contribution" or "connection strength" of an element within a set. A system is called monotone if, as the set expands, the "connectedness" of an element within it does not increase (or does not decrease, depending on the type of system). The theory proves that for any such system, there exists a unique sequence of nested subsets, each of which is maximally connected for its level. Unlike many NP-hard clustering problems, core search in a monotone system is performed in polynomial time using a simple greedy algorithm (sequentially removing the "weakest" element).

Mirkin (2012) proposed an original theoretical approach that views clustering as a process of reconstructing the hidden structure of data. This approach allows for the unification of disparate methods, such as K-means and Ward's hierarchical clustering, into a unified mathematical framework. In (Mullat, 2004) a monotone system is considered as a set of subsets, the elements of which possess certain "indicators" or "credentials". These indicators change monotonically when the composition of the subsets changes. A procedure for identifying the most significant subsystems—cores—is defined. Positive and negative cores represent elements that are most sensitive to changes ("actions") within the system. Removing such a core entails a radical change in the properties of the entire structure.

Muchnik and Shvartser (2010) focus on discrete dynamical systems where state transitions occur stepwise, which is critical for problems of classification, pattern recognition, and data analysis. The methods described in the paper find application in combinatorial data analysis (CDA), bioinformatics (e.g., for gene clustering), and social choice theory. The authors demonstrate that using local properties allows one to construct more efficient algorithms for finding extremal subsystems and kernels of monotone systems, which simplifies the processing of large data sets.

In (Mullat and Leetma, 2016) the authors present algorithms for reordering data tables in order to give them a more informative form. The main goal of the algorithms is to summarize the entire data table using one "average" object, called the "best decision". This object is defined as the one that gives the maximum value of the weight function. All the described methods are based on

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1005 DOI: 10.66693/mathai.1005
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

the mathematical apparatus of monotone systems, which allows one to analyze the dynamics of indicators that increase or decrease in accordance with a partial order in subsets.

Mullat (2024) present an updated version of fundamental research on the theory of extremal subsystems. The revised version focuses on the formulation and proof of a duality theorem, which describes a systematic approach to restricting the search space for kernels and stable sets in monotonic systems. A constructive algorithm for finding extremal subsystems, implemented as a dual scheme, is presented. The methodology is used to analyze data structures, graphs, and networks, allowing one to extract highly cohesive or self-contained units within complex systems.

In his study Mirkin (2007) adapts the rigorous mathematical framework of monotonic systems to solve applied problems in information systems and management. This publication is significant in that it transfers the theoretical developments of the 1970s and 1980s into the context of modern data mining, making them accessible to business analytics professionals.

The work of Yao and Liu (2024) finds the most densely connected groups of users, ignoring "noise" and the periphery (community detection). In the search for groups of genes that behave similarly under certain conditions, monotonic systems make it possible to find stable gene clusters even in the presence of strong noise in the sequencing data (Datta and Datta, 2006). Parupudi (2025) implement feature selection to identify the most informative variables that maintain a monotonic relationship with the target variable using the framework of monotonic systems.

Makdesi et al. (2023) propose a model mapping that provably contains all monotonic functions consistent with the data. This mapping is also minimal in the sense that any set-valued mapping containing all consistent monotonic functions will also include the proposed one. It is shown that this minimal model mapping is interval-valued and allows for a simple construction on a finite partition induced by the data. Since the complexity of the partition increases with the data volume, the authors consider the problem of computing minimal interval-valued model mappings defined on a priori fixed partitions. Finally, algorithms are proposed for constructing guaranteed approximations of unknown monotonic functions based on a discrete data set.

An interesting approach to constructing barrier certificates for unknown monotone systems using a finite number of system trajectories is presented, while providing formal correctness guarantees (Galarza-Jimenez et al., 2025). Given several system trajectories, a family of monotone basis functions is constructed that remain nonincreasing on all trajectories. Using this class of basic functions, sample-based optimization is performed to construct barrier certificates, ensuring system security without the need for additional modeled data or the assumption of Lipschitz continuity.

Thus, the analysis of publications shows that the theory of monotone systems, the foundations of which were formulated by the school of I. Mullat, I. Muchnik and B. Mirkin, presents great potential for its development, which is the subject of this theoretical study. The purpose of this article, taking into account the above-described properties of monotone systems, is to formulate the statement of the problem of specifying in a monotone system a set of basic functions of three types in the class of algorithms for calculating estimates (ECA) (Zhuravlev et al., 1974).

3 METHODOLOGY

The theory of monotone systems (MS) identifies a new class of effectively solvable combinatorial-extremal problems and, in this sense, has a broader significance than simply another theory of methods for processing empirical information. An interesting application of the theory of monotone systems is its application to structural methods of data analysis. One of the main elements of this analysis is the identification of sub-matrices that formalize the original structure in the form of an "object-feature" matrix (Braverman and Muchnik, 1983). Such a formalization requires the construction of a flexible mathematical framework with a set of transition operators for moving between different formalizations. This will allow one to "select" the appropriate formal procedure for extracting submatrices and construct a detailed description of the system. This is the idea behind the monotone system generator, as a system for generating a family of monotone systems. In general, the MS generator is represented by a two-stage implementation. In the first, a set of operators for transforming monotone systems of a fairly general form is constructed, and in the second stage, a set of basic functions is designed.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1005 DOI: 10.66693/mathai.1005
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

3.1 MONOTONIC SYSTEMS

Below, we briefly outline the key principles of monotonic systems theory as presented in Mullat (1977). Let W be a finite set of elements, $|W| = N$, and let π be a scalar function which assigns to each pair (i, H), where $H \subseteq W$ is an arbitrary subset of W and $i \in H$, a number $\pi(i, H)$, where the meaning of $\pi(i, H)$ is the affinity of the element i to the subset H. $\pi(i, H)$ can also be interpreted as the importance of element i in subset H.

For example, if a matrix $\|\rho_{ij}\|_N^N$ of distances between all pairs of elements is given for a set W, then the simplest function $\pi(i, H)$ is the sum of the distances from an element to all other elements:

$$\pi = \sum_{j \in H} \rho_{ij}. \tag{1}$$

Then the system $\langle W, \pi \rangle$, consisting of a finite set W with a set of pairs (i, H) given on it and the function $\pi(i, H)$, is called *monotone* if:

$$\pi(i, H \setminus j) \leq \pi(i, H), \quad \forall\, i, j \in H,\ i \neq j,\ \forall H \subseteq W, \tag{2}$$

or

$$\pi(i, H \setminus j) \geq \pi(i, H), \quad \forall\, i, j \in H,\ i \neq j,\ \forall H \subseteq W. \tag{3}$$

If inequality (2) is satisfied, the system is called $(-)$-monotone and is denoted by (W, π^-), and if condition (3) is satisfied, then $(+)$-monotone and is denoted by (W, π^+).

The main goal of MS theory is to identify some extremal subsystem of it, which is called the largest core. On the set W of all subsets of the $(-)$-monotone system, a scalar function $F^-(H)$ is defined, which is assigned to each subset $H \subseteq W$:

$$F^-(H) = \min_{i \in H} \pi^-(i, H), \quad \forall H \subseteq W. \tag{4}$$

Definition 1. The *kernels* of a $(-)$-monotone system $\langle W, \pi \rangle$ are those subsets of the set W on which the maximum of the function $F^-(H)$ is achieved.

$$F^+(H) = \max_{i \in H} \pi^+(i, H), \quad \forall H \subseteq W. \tag{5}$$

Let the elements of the set W be ordered arbitrarily. The resulting sequence of elements $A = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$, where $W = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$, one-to-one corresponds to the sequence $\bar{H}(A)$, or $\bar{H} = \langle H_1, \ldots, H_N \rangle$ of nested subsets of the set W, where $H_1 = W$; $H_2 = H_1 \setminus \alpha_1$; $\ldots$; $H_{k+1} = H_k \setminus \alpha_k$; $\ldots$; $H_N = \{\alpha_N\}$.

Definition 2. An ordered sequence A of elements of a set W is called a *defining sequence* $(-)$ of a monotone system $\langle W, \pi \rangle$ if in the corresponding sequence of sets $\bar{H}$ there exists a sequence $\bar{\Gamma} = \langle \Gamma_1, \ldots, \Gamma_P \rangle$, where $\Gamma_1 = H_1 = W$, such that

$$\begin{aligned} &\pi(\alpha_k, H_k) < F(\Gamma_{j+1}), \quad \forall\, \alpha_k \in \Gamma_j \setminus \Gamma_{j+1},\ j = \overline{1, p-1}, \\ &F(L) \leq F(\Gamma_P), \quad \forall\, L \subset \Gamma_P. \end{aligned} \tag{6}$$

Definition 3. A set $G \subseteq W$ is called a *definable set* of a monotone system $\langle W, \pi \rangle$ if there exists a defining sequence such that $\Gamma_P = G$.

Central theorems play an important role in the theory of MS. The first of these asserts the attainability of the global maximum of the function $F^-(H)$ on the definable set of a $(-)$-monotone system and the fact that all kernels of the $(-)$-MS lie within a defined set. The second theorem establishes the closure of the system of all sets X on W with respect to the binary operation of union of the sets on which the function F^- attains the global maximum.

Thus, the core of the MS is the last set $G = \Gamma_P$ in the sequence $\bar{\Gamma}$ obtained by constructing the defining sequence. Interest in monotonic systems is currently growing due to the problem of Big Data and the increasing demands on the interpretation of the results of structural data analysis. Monotonic systems, as a tool for associative-structural analysis, represent an attempt to formalize analogous processes occurring in the human brain.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1005 DOI: 10.66693/mathai.1005
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

3.2 Assigning Similarity in ECA

Estimating the similarity of object parts is the central element of the ECA method. A subset of features ω ($\omega \subseteq P$) is selected from the set of features. In the feature-object matrix, the set S' of object coordinates corresponding to the subset ω is called the ω-part of the object. A binary indistinguishability relation $R_\omega(S, S')$ is defined on the set of ω-parts of all objects under consideration. The target similarity function $r_\omega(S, S')$ between S and S' is given by:

$$r_\omega(S, S') = \begin{cases} 1, & \text{if there is a relation } R_\omega(S, S') \text{ between } \omega\text{-parts } S \text{ and } S', \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

A family of feature subsets Ω is identified, and a relation $R_\omega(S, S')$ is defined on each element $\omega \in \Omega$. Then, each pair of objects S and S' divides the family Ω into two disjoint subfamilies $\Omega^0(S, S')$ and $\Omega'(S, S')$:

$$\begin{aligned} &\text{a) } \omega \in \Omega^0(S, S')\,, \text{ if } r_\omega(S, S') = 0, \\ &\text{b) } \omega \in \Omega'(S, S')\,, \text{ if } r_\omega(S, S') = 1, \end{aligned} \tag{8}$$

so that $\Omega = \Omega^0(S, S') \cup \Omega'(S, S')$.

As a function of the similarity of a pair of objects in the ECA, we select $|\Omega'(S, S')|$, which is equal to the number of $r_\omega(S, S') = 1$ on the pair (S, S'), written in the form:

$$f(S, S') = \sum_{\omega \in \Omega} r_\omega(S, S')\,. \tag{9}$$

From (8), the flexibility of the ECA becomes apparent, which is ensured by the wide possibilities of changing the family Ω and relations $R_\omega(S, S')$ and the ability to include a priori information about the analyzed objects in the selection of the family Ω and in the definition of $R_\omega(S, S')$ on it.

In the case of binary features, the set Ω_n of all single features is written as:

$$\Omega_n = P_1 : |\Omega_n| = |P| = n.$$

If Ω_q is the set of all possible subsets of P, then

$$r_\omega(S, S') = \begin{cases} 1, & \text{if } \omega\text{-parts } S \text{ and } S' \text{ coincide,} \\ 0, & \text{otherwise.} \end{cases}$$

The similarity functions in the case of binary features f_1 and in the case Ω_q will have the form:

$$f_1(S, S') = n - \rho(S, S')\,, \tag{10}$$

$$f_2(S, S') = 2^{n - \rho(S, S')}, \tag{11}$$

where $\rho(S, S')$ is the Hamming distance between S and S'.

Between a linear function f_1 and a nonlinear function f_2, there is a whole range of functions of intermediate complexity. As a family of support sets Ω, all subsets whose cardinality does not exceed a certain given value k ($k \leq n$) can be chosen:

$$\Omega_{np} = \bigcup_{i=1}^{k} \Omega_i, \tag{12}$$

where Ω_i is the family of all subsets of features of power i, and $r_\omega(S, S')$ for each $\omega \in \Omega_{np}$ is calculated according to (7). Then, the similarity function $f_{np}(S, S')$ takes the form:

$$f_{np}(S, S') = \sum_{i=1}^{\tilde{k}} C^i_{n - \rho_i(S, S')}, \tag{13}$$

where $\tilde{k}$ is determined from the equation:

$$\tilde{k} = \begin{cases} k, & \text{if } k \leq n - \rho(S, S')\,, \\ n - \rho, & \text{if } k \geq n - \rho(S, S')\,. \end{cases}$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1005 DOI: 10.66693/mathai.1005
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

The function $f_{np}(S, S')$ is not complex, as it does not require explicit enumeration of the family Ω of support sets.

Conditions (7) can be relaxed by introducing a threshold ε_ω:

$$r_\omega(S, S') = \begin{cases} 1, & \text{if } \rho_\omega(S, S') < \varepsilon_\omega, \\ 0, & \text{otherwise.} \end{cases}$$

It is clear that the introduction of ε_ω gives a free parameter of the function $f_{np}(S, S')$.

3.3 Defining the Base Functions of Monotone Systems in the ECA Class

We will consider three types of functions $\pi(i, H)$. The *first type* is a weight function of pairwise connections between an element $i \in H$ and each element of this set:

$$\pi(i, H) = \sum_{k \in H} a_{ik} = \sum_{k \in H} \sum_{\omega \in \Omega} r_\omega(i, k), \quad i \in H,\ H \subseteq W, \tag{14}$$

where $r_\omega(i, k)$ is the similarity function of elements i and k.

The *second type* of weighting function differs from the first in that it introduces a "pattern" on the set H, which can be selected from H in some way. For example, such a pattern could be some object i for which the majority of objects in this subset "vote", the cardinality of which $|H| = N_H$:

$$\Gamma_H(i) = \max_{j \in 1, N_H} \{\Gamma_1(i), \Gamma_2(i), \ldots, \Gamma_j(i)\}. \tag{15}$$

The number of votes $\Gamma_j(i)$ is determined on the set of ω-parts of all objects under consideration:

$$\Gamma_j(i) = \frac{1}{|H| - 1} \sum_{i,j \in H} f(i, j), \tag{16}$$

where $f(i, j) = \sum_{\omega \in \Omega} r_\omega(i, j)$ is the affinity function of i and j.

Interestingly, a special "object" or "anti-pattern" can be used as a representative of a set H. Formally, the problem of identifying special objects is represented as a partial approximation problem. Then, a representative of a set H can serve as a pattern or anti-pattern for the weight functions of the second-type MS.

Weight functions of the *third type* measure the relationship between an element and a subset as the sum of the relationships between the element and the group for each individual feature. Each feature j is characterized by a certain weight $\omega_j(H)$, which depends on the composition of the set H (j-th feature weight). Obviously, $\omega_j(H)$ must be monotonic:

$$\omega_j(E) \leq \omega_j(H), \quad \forall E \subseteq H \subseteq W,\ \forall j \in Y.$$

In general, the measure of importance of the l-th feature $P(l)$, which is understood as its ability to "separate" an object from a certain set, is calculated as:

$$P(l) = \frac{\sum\limits_{t=1}^{m-1} \sum\limits_{j=t+1}^{m} C^k_{r(S_t, S_j)} - \sum\limits_{t=1}^{m-1} \sum\limits_{j=t+1}^{m} C^k_{r(S_t, S_j) \setminus l}}{\sum\limits_{t=1}^{m-1} \sum\limits_{j=t+1}^{m} C^k_{r(S_t, S_j)}}, \tag{17}$$

where $C^k_{r(S_t, S_j)}$ is the measure of the similarity of a pair of rows S_t, S_j on a supporting subset of the cardinality k in the case of the deletion of the l-th feature. The cardinality of the subset k is an important parameter, and its optimization methods are the subject of future research in the context of MS.

The measure of the importance of the l-th feature of the object $S_t \in H$:

$$P(l)_{S_t} = \frac{\sum\limits_{j=1}^{|H|} C_r(S_t, S_j) - \sum\limits_{j=1}^{|H|} C_r(S_t, S_j)_{\setminus l}}{\sum\limits_{j=1}^{|H|} C_r(S_t, S_j)}, \quad j, t = \overline{1, |H|},\ j \neq t. \tag{18}$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1005 DOI: 10.66693/mathai.1005
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Then the relation function $\pi(S_t, H)$ of the element S_t and the subset H:

$$\pi(S_t, H) = \sum_{l=1}^{n} P(l)_{S_t} \tag{19}$$

shows the distance of S_t from the subset H by all features, weighted in accordance with (18).

There is a known option for determining the weight of a feature in the process of constructing a defining sequence of a monotonic system, which leads to the standard form of a linear programming problem with respect to an unknown system of weights (Eshonkulov, 1993). This study develops the classical direction of Yu. I. Zhuravlev's pattern recognition, adapting it to the tasks of taxonomy (clusterization). A distinctive feature is adaptive methods for adjusting algorithm parameters directly to a specific data sample, which improves the accuracy of classification and identification of significant features.

4 Conclusion

Defining a system of base functions for monotonic systems within the class of Estimate Calculation Algorithms (ECA) provides a significant advantage in constructing efficient procedures for identifying the core of a monotonic system, as well as in enhancing the interpretability of the resulting solutions. The presented formalization of constructing base functions within the ECA framework remains insufficiently explored and is, therefore, a subject for future research, including experimental and comparative studies of the proposed approach.

Author Contributions

F.T. Adilova conceived the research idea and developed the theoretical framework. R.R. Davronov contributed to the formalization and manuscript preparation.

Acknowledgments

The authors gratefully acknowledge the support of the V.I. Romanovskiy Institute of Mathematics, Uzbekistan Academy of Sciences.

References

T. Aaremaa, Aaremaa, T. (1986). Monotone systems with several kernel-subsets. *Information Processing Letters*, 21(2).

(1983). *Structural Methods for Processing Empirical Data*. Science.

S. Datta and S. Datta, Datta, S. and Datta, S. (2006). Methods for evaluating clustering algorithms for gene expression data using a reference set of functional classes. *BMC Bioinformatics*.

T. Eshonkulov, Eshonkulov, T. (1993). *Adaptive algorithms for calculating estimates in problems of taxonomy and feature extraction*. PhD thesis, Tashkent. Abstract of PhD dissertation.

F. Galarza-Jimenez, M. Zamani, and S. Jafarpour, Galarza-Jimenez, F. and Zamani, M. and Jafarpour, S. (2025). Trajectory-based barrier certificates for monotone systems. pp., 5831–5836, Rio de Janeiro, Brazil.

E. N. Kuznetsov and I. B. Muchnik, Kuznetsov, E. N. and Muchnik, I. B. (1982). Analysis of the distribution of functions in an organizational system. *Automation and Telemechanics*, (10):119–127.

B. Makdesi, A. Girard, and T. Bastogne, Makdesi, B. and Girard, A. and Bastogne, T. (2023). Data-driven models of monotone systems. *IEEE Transactions on Automatic Control*.

B. Mirkin, Mirkin, B. (2007). Monotone systems and data analysis methods. pp., 33–42.

(2012). *Clustering: Principles and Practice*. CRC Press (Chapman & Hall).

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1005 DOI: 10.66693/mathai.1005
Special Issue: MathAI 2026 Selected Papers
Received: 31 Mar 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

B. Mirkin and I. Muchnik, Mirkin, B. and Muchnik, I. (2002). Layered clusters of tightness set functions. *Applied Mathematics Letters*, 15:147–151.

I. Muchnik and I. Shvartser, Muchnik, I. and Shvartser, I. (2010). Discrete monotone systems and locality. *Lobachevskii Journal of Mathematics*, 31(4):343–353.

D. Mullat, Mullat, D. (2004). Monotone systems in structural analysis of data. Monotone Phenomena of Issues Behind Bargaining Games and Data Analysis. Technical report. Updated 2024.

I. E. Mullat, Mullat, I. E. (1976). Extreme subsystems of monotonic systems. *Automation and Telemechanics*, (5):130–139.

I. E. Mullat, Mullat, I. E. (1977). Extremal subsystems of monotonic systems, I, II, III. *Automation and Remote Control*, 37. Part I: vol. 37(5), pp. 758–766; Part II: vol. 37(6), pp. 920–929; Part III: vol. 38(1), pp. 52–66.

J. Mullat and M. Leetma, Mullat, J. and Leetma, M. (2016). Some monotone systems algorithms for data mining. *International Journal of Geology*, 10:101–110.

Joseph Mullat, Mullat, Joseph (2024). Extremal subsystems of monotonic systems, I–II (revised 2024). Monotone Phenomena of Issues Behind Bargaining Games and Data Analysis. Technical report, MBS Nederland.

V. S. Raghu Parupudi, Parupudi, V. S. Raghu (2025). Magnitude matters: a superior class of similarity metrics for holistic semantic understanding. arXiv:2509.19323 v.1.

J. Yao and B. Liu, Yao, J. and Liu, B. (2024). Community-detection method of complex network based on node influence analysis. *Symmetry*, 16:754.

(1974). *Algorithms for Calculating Estimates and Their Application*. FAN, Tashkent.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

DEEP LEARNING FOR EDUCATIONAL VIDEO ANALYSIS: BENCHMARKING ASR SYSTEMS AND PIPELINE OPTIMIZATION

Gordey Zuev
Faculty of Computer Science
HSE University
Moscow, Russia

Elena Kantonistova
Faculty of Computer Science
HSE University
Moscow, Russia

ABSTRACT

We present a comparative analysis of eight *managed* commercial speech recognition providers (provider-side preprocessing, segmentation, and serving) for educational video transcription and enrichment, evaluated on over 700 lecture recordings (900+ hours) across disciplines. The Fireworks `whisper-v3-turbo` endpoint offers a favorable cost-quality-latency trade-off versus surveyed alternatives. Audio preprocessing reduces billed duration by 10-25% with negligible accuracy loss. Prompt-based "Video Vocabulary" reduces terminology errors without fine-tuning. We implement a parallel pipeline that cuts end-to-end turnaround from over 30 minutes of manual effort per recording to under two minutes, supports up to 50 concurrent jobs, and achieves $\sim 22\times$ speedup at $\approx$ \$0.075 per hour of content for transcription plus pedagogical enrichment (summaries, chapter topics, self-check questions) at list prices. The system is deployed in production.

1 INTRODUCTION

The widespread transition of educational institutions to hybrid and distance learning has led to a substantial increase in recorded lectures (Dhawan, 2020). Universities generate hundreds of hours of video content monthly, mirroring the scale of initiatives that successfully archived over 1,000 lectures annually even two decades ago (Nagataki et al., 2006). While modern conferencing tools have automated recording itself, post-production – trimming, metadata, timestamps, subtitles – remains manual. Based on our experience, this requires at least 30 minutes per recording. This aligns with conservative manual production estimates, which recent work suggests can be reduced by two orders of magnitude through automation (Holmberg, 2025). With 100 recordings monthly, manual effort exceeds 50 person-hours, a burden underscoring the need for automation, particularly as prior university efforts showed even basic recording required significant human resources (e.g., 2 hours per lecture) (Nagataki et al., 2006).

Existing solutions fall into two categories: transcription services and low-code automation platforms. Our analysis of 15 solutions revealed none provide a fully automated processing cycle, while transcription costs vary by more than 60-fold–from roughly \$ 0.06 to \$ 3.8 per hour of recording (order-of-magnitude market range, converted to USD at 79 RUB/USD where vendors quote in RUB).

While prior work has focused either on improving ASR accuracy for educational content (Tobias, 2025; Rao, 2023) or on pipeline automation using commercial tools (Holmberg, 2025), no existing study systematically addresses the full spectrum from cost-optimized transcription to pedagogical content enrichment. We bridge this gap with the first end-to-end platform that enables any educational institution or organization producing video at scale to centrally upload recordings, automatically process them (transcription, trimming, enrichment), and publish to multiple destinations with a single action or on a configured schedule. We demonstrate that production-ready scalability need not compromise on either accuracy or downstream functionality – including automatic summary generation and open-ended question creation, features absent from previous educational video platforms.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

This paper makes the following key contributions:

- *Benchmark.* Systematic survey of the commercial APIs in Table 1 for Russian and English lecture speech, identifying favorable cost-latency-feature trade-offs under production constraints (not a controlled ablation of isolated model weights).
- *Optimization.* Preprocessing pipeline (silence suppression, noise gating, compression) yielding 10-25% cost reduction.
- *Domain adaptation.* Evaluation of prompt-based Video Vocabulary mechanism that reduces terminology errors without expensive fine-tuning.
- *System.* Production-ready parallel processing platform supporting up to 50 concurrent recordings.
- *Enrichment.* Automated generation of lecture summaries and open-ended questions at marginal additional cost, transforming transcripts into pedagogical resources.
- *User feedback.* Post-deployment questionnaire (n=237) with multiple Likert items on topics, timestamps, summaries, and self-check questions; exploratory descriptive evidence of acceptance (not an RCT).

To guide our investigation, we formulate three research questions:

1. How do transcription APIs compare in cost and quality for Russian lecture content?
2. What preprocessing techniques reduce costs without compromising accuracy?
3. What architecture enables efficient parallel video processing at scale?

All quantitative estimates – processing time, speedup factor, API costs, labor savings – were obtained experimentally on a corpus of over 700 real video recordings (totaling over 900 hours of content) and averaged over the full corpus; methodology is described in Section 5. Our main empirical results are formalized as Empirical Results 1, 2, and 3, and Observation 1. Code and documentation are available at the project repository LEAP (2026).

2 Related Work

Accented and non-native speech remains a known structural challenge for ASR relative to mainstream reference conditions (Hinsvark et al., 2021); we return to speaker effects in Section 6. On the systems side, our deployed platform uses a layered service decomposition with swappable integrations (Fowler, 2002), developed in Section 4.

2.1 Transcription Services and Automation Platforms

Existing solutions can be categorized into two main groups.

(i) Transcription services and APIs (Fireworks AI (Fireworks AI, 2026), OpenAI Whisper (OpenAI, 2026), Google Cloud STT (Google Cloud, 2026), AssemblyAI (AssemblyAI, 2026), Gladia (Gladia, 2026), Deepgram (Deepgram, 2026), ElevenLabs (ElevenLabs, 2026)) address only the speech-to-text conversion stage. Several providers impose file size limitations: OpenAI (25 MB; typical lecture video $\sim$500 MB), Memo AI (0.5-2 GB depending on the plan). No service provides topic extraction with timestamps.

(ii) Low-code automation platforms (Make.com, Zapier, n8n) enable pipeline construction from API calls but lack dedicated CPU workers for video-stream processing (trimming), impose file size limitations (100–500 MB), and require substantial setup and maintenance effort (roughly $ 380–$ 760 one-time development and $ 63–$ 190/month maintenance in our market survey, converted at 79 RUB/USD). Users must additionally provision server infrastructure for video processing.

2.2 Whisper for Educational Content

Recent research has specifically examined the application of OpenAI's Whisper models to educational video transcription. Rao (2023) conducted a preliminary study on 25 academic videos,

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

demonstrating that even the larger Whisper models transcribe content in less than 25% of the playback time – a $4\times$ speedup over human transcription. However, the study also identified key challenges: Whisper tends to generate plausible but incorrect text on inaudible segments, which inflates Word Error Rate (WER) metrics when compared against human transcripts that mark such segments as inaudible.

Tobias (2025) addressed these limitations through domain-specific fine-tuning. By adapting Whisper-small on lecture recordings (including a self-curated 10-hour dataset), the optimized model achieved a WER of 4.53% on unseen educational audio – surpassing Whisper-Medium (5.51%) and Whisper-Large-v2 (5.78%). This work demonstrates that domain adaptation can significantly enhance ASR accuracy for educational content. However, fine-tuning remains computationally expensive and requires curated datasets. Our work explores a lighter alternative: we pair the Whisper architecture family (Radford et al., 2023) with preprocessing optimizations and evaluate prompt-based adaptation via domain-specific vocabulary injection through a *managed* Whisper-compatible API (Fireworks AI's `whisper-v3-turbo` endpoint), achieving accuracy gains at zero training cost.

We deliberately focus on cloud-based API solutions rather than self-hosted checkpoints. Managed endpoints offer low marginal cost per hour of audio and documented batch RTF far below interactive needs, without operating our own GPU fleet for the acoustic model. Self-hosting open weights at our peak load (50+ concurrent recordings) would imply substantial capital and operations cost. Table 2 still matters for orientation: it summarizes RTF ranges, language coverage, and prompting constraints for future model choices. Our platform keeps ASR and LLM calls behind stable orchestration so that swapping providers is mostly configuration work.

The pipeline serves a mixed Russian-English catalog (50+ hours of English processed so far). Multilingual Whisper-style APIs cover both in one integration. Many Russian-centric checkpoints ship as separate per-language models (e.g., Vosk (Vosk, 2026)), which complicates a single uniform stack. Accented and non-native speech also remains structurally harder for ASR than mainstream reference conditions, as summarized above. These are engineering-economic criteria for our benchmark. They do not imply superiority of one open-weight architecture over another outside this deployment context.

2.3 Russian-Language and Optimized ASR Systems

Several open-source Russian-language ASR systems exist, including GigaAM (Sber) (Salute Developers, 2025), T-one (T-Bank) (T-Bank, 2025), Vosk (Vosk, 2026), and community fine-tunes such as bond005/whisper-podlodka-turbo (Bondarenko, 2025) and Vikhrmodels/Borealis (Kuleshov and Nikolich, 2025). Table 2 summarizes representative characteristics. These systems are *not* omitted because they are inherently inferior. Our primary empirical comparison targets managed ASR APIs (fixed per-minute pricing, provider-operated preprocessing, elastic scaling). Self-hosted stacks entail separate hardware sizing, batching policies, and RTF that are not uniquely determined without a dedicated controlled study on our corpus. Video Vocabulary requires a `prompt`-compatible decoder path with whole-request biasing on long recordings. Many Russian checkpoints (GigaAM, T-one, Vosk, Pisets) do not expose an equivalent mechanism in the same form. WhisperX applies the initial prompt only to the first window of the pass (Bain et al., 2023), which is suboptimal for hour-long vocabulary-sensitive lectures. Community Whisper fine-tunes (e.g., podlodka) *do* support prompts but still imply self-hosted serving and monitoring.

Recent work has also produced optimized Whisper-based pipelines: Faster-Whisper (SYSTRAN, 2026) (CTranslate2 port of Whisper reporting large wall-clock reductions versus reference PyTorch inference on GPU under documented conditions (SYSTRAN, 2026)), WhisperX (Bain et al., 2023) adding phoneme-level alignment and VAD-based segmentation, and Pisets (Bondarenko et al., 2025) designed specifically for lecture robustness. These are important engineering references; they are cited here for completeness and to situate our API-centric benchmark. Their RTF varies with hardware and batching; for example, Faster-Whisper documents RTF roughly 0.02-0.08 on an RTX 3070 Ti for large-v2 under specific settings (SYSTRAN, 2026). For our production setting we prioritized managed APIs for predictable cost, multilingual coverage in one vendor integration, and operational simplicity, while acknowledging that a future controlled on-premise evaluation could quantify head-to-head WER under matched audio and hardware.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

3 Evaluation of Commercial ASR APIs and Self-Hosted Alternatives

3.1 Transcription System Benchmarking

Following common practice in machine-learning-as-a-service evaluation, we distinguish (i) an ASR model in the narrow research sense – a reproducible checkpoint with explicit architecture and weights, under user-controlled pre/post-processing – from (ii) a managed ASR API, i.e., a provider-operated *system* that bundles one or more acoustic models with segmentation, Voice Activity Detection (VAD), resampling, and undisclosed or versioned server-side logic. Our primary benchmark compares items of type (ii) using published tariffs, documented latency/throughput where available, and prompting features. We do not claim to isolate the contribution of the acoustic network from provider infrastructure; self-hosted rows in Table 2 are contextual references, not paired measurements on our 700+ recordings unless stated otherwise.

We surveyed commercial transcription API providers along cost, latency, and feature criteria for Russian and English speech recognition. Tables 1 and 2 summarize commercial API services and representative self-hosted alternatives, respectively.

Table 1: Commercial ASR API services (March 2026).

Provider	Advertised backend	Price ($/min)	RTF / latency	Prompting	Languages
Fireworks AI	Whisper-v3-turbo	0.0009 (Fireworks AI, 2026)	0.00083–0.002 (Fireworks AI, 2026)	`prompt`	99
Fireworks AI	Whisper-v3-large	0.0015 (Fireworks AI, 2026)	0.00125–0.003 (Fireworks AI, 2026)	`prompt`	99
OpenAI	Whisper API	0.006 (OpenAI, 2026)	0.03 (Fireworks AI, 2026)	`prompt`	99
OpenAI	GPT-4o Mini Transcribe	0.003 (OpenAI, 2026)	0.02–0.05	`prompt`	Multilingual
AssemblyAI	Universal-3 Pro	0.0035 (AssemblyAI, 2026)	0.02–0.05	`prompt`, `keyterms` (AssemblyAI, 2026)	6
Deepgram	Nova-3	0.0077 (Deepgram, 2026)	TTFT 200–300 ms (TalkFlow AI, 2025)	`keyterm` (Deepgram, 2026)	36+
Deepgram	Nova-3 (Cloudflare)	0.0052 (Cloudflare, 2026)	TTFT ∼300 ms	`keyterm`	10+ (incl. RU)
Gladia	Async / Real-time	0.006–0.0125 (Gladia, 2026)	TTFT <300 ms (Gladia, 2026)	Custom vocab (beta)	100+
ElevenLabs	Scribe v1	0.0037–0.007 (ElevenLabs, 2026)	0.03–0.1	Keyterms, entities (ElevenLabs, 2026)	90+
Azure	Batch STT	0.006	0.1–0.3	Phrase List	100+
Google Cloud	STT v2	0.016 / 0.003 (batch)	0.05–0.15	PhraseSet (Google Cloud, 2026)	125+

RTF and latency are from provider documentation (not measured on our corpus); Fireworks batch RTF is detailed in Fireworks AI (2026). Uncited cells are indicative. *TTFT* denotes streaming latency to the first transcript output (not LLM text generation) and is not comparable to batch RTF.

Table 2: Representative self-hosted ASR systems (checkpoint + stack). RU WER ranges are *illustrative* and not comparable across rows. Reported hardware for RTF benchmarks is in Table 4.

System	Architecture	RTF	Prompting	Languages	RU WER
GigaAM-v3	Conformer CTC/RNNT	0.05–0.15 (Salute Developers, 2025)	None	Russian	3–10% (Salute Developers, 2025)
T-one	CTC streaming Conformer	<0.01 (stream) (T-Bank, 2025)	None	Russian	6–9%
Vosk	Kaldi	0.2–0.5 (Vosk, 2020)	None	Russian	∼11% (Alphacephei, 2025)
podlodka-turbo	Whisper fine-tune	0.03–0.1 (Bondarenko, 2025)	`prompt`	Russian, English	9–14% (Alphacephei, 2025)
Borealis	Audio LLM (Whisper+Qwen)	0.05–0.2 (Kuleshov and Nikolich, 2025)	LLM instruction	Russian, English	6–16% (Alphacephei, 2025)
Faster-Whisper	CTranslate2 (Whisper)	0.02–0.08 (SYSTRAN, 2026)	`prompt`	99	∼10%
WhisperX	Whisper + alignment	0.1–0.3 (Bain et al., 2023)	`prompt` (init. only)	99+	∼10%
Pisets	Wav2Vec2 + AST + Whisper	0.15–0.25 / 1.0–1.5 (Bondarenko et al., 2025)	None	Russian, English	2–5% (Bondarenko et al., 2025)

For Vosk, "Russian" denotes that vendor's Russian acoustic model; the toolkit supports many languages via separate packs (Vosk, 2026). Encoder-only RTF for GigaAM is indicative of full-stack latency (Salute Developers, 2025).

Among the APIs surveyed in Table 1, the Fireworks AI endpoint advertising `whisper-v3-turbo` offered the lowest published per-minute tariff for our workload: $0.0009/min (Fireworks AI, 2026), i.e. $ 0.054/h of audio at list price–roughly 2.8× to 18× lower than several alternatives at similar documented batch RTF tiers. We emphasize price, latency, and feature fit. We do not report a fully controlled WER ranking across all surveyed providers on a shared reference transcript set in this paper. In operational use on our lecture corpus, transcription quality from this endpoint was sufficient for downstream summarization and subtitles. Isolating acoustic WER from provider-side normalization would require additional controlled experiments.

The choice of `fireworks/whisper-v3-turbo` is justified by the provider's architectural optimization for throughput: a pruned decoder relative to Whisper-large-v3 (decoder layers reduced from 32 to 4 with encoder retained), as described in Fireworks documentation (Fireworks AI, 2026), following decoder-compression ideas related to knowledge distillation (Gandhi et al., 2023). The provider reports large end-to-end speedups versus full-decoder Whisper on the same serving stack

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

(Fireworks AI, 2026); these factors should be interpreted as service-level performance, not as a claim about the open Whisper checkpoint alone. The Fireworks API additionally provides features we rely on in production: built-in VAD for silence handling and chunking of long audio into the model's receptive field with stitched outputs (Fireworks AI, 2026). For Video Vocabulary, Fireworks documents whole-request `prompt` conditioning so that injected terms can influence recognition across an entire uploaded recording (Fireworks AI, 2026). Optimized local stacks such as WhisperX target time-accurate alignment and long-form transcription workflows (Bain et al., 2023); they are cited in Section 2 as related engineering rather than as APIs included in our tariff survey.

Scope and limitations. Our API comparison does *not* separate the effect of neural acoustic weights from proprietary preprocessing, chunking, or server versioning. A rigorous isolation would require local deployment of fixed checkpoints with matched front-end processing. We state this limitation explicitly because it affects the interpretation of "model" claims in cloud settings.

3.2 Cost Optimization Through Audio Preprocessing

Transcription cost via API scales linearly with audio duration. We evaluated three optimization methods:

- *Audio extraction.* Using FFmpeg (FFmpeg, 2026) to extract audio from video reduces data transfer from approximately 700 MB to 50 MB and circumvents OpenAI's 25 MB file limitation.
- *Silence removal.* Automated detection and removal of pauses reduces billable duration. Lecture recordings frequently contain extended periods of silence – waiting for an audience to settle before beginning, long pauses between topics, or delays at the end before stopping the recording. Our pipeline automatically detects and strips these segments, reducing effective duration by 10-25% and delivering substantial savings when processing large volumes of content.
- *Signal enhancement.* Amplitude normalization and dynamic range compression increase speech contrast relative to background noise, improving recognition accuracy on low-quality recordings. Noise reduction techniques including spectral gating were applied to mitigate background interference.

3.3 Domain Adaptation via Video Vocabulary

To address the challenge of domain-specific terminology without resorting to computationally expensive fine-tuning, we evaluate a prompt-based approach. For each recording, we define a set of 5-10 key terms, proper names, or specialized lexicon. This vocabulary is passed as a prompt to the Whisper-v3-turbo model via the API (Fireworks AI, 2026). The model's decoder uses this prompt as a contextual guide, biasing the output distribution toward the provided terms. Crucially, the Fireworks API applies the prompt globally across the entire audio (whole-request scope), ensuring that terminology guidance persists throughout long recordings.

This prompt-based approach offers several advantages over fine-tuning:

- *Zero computational cost:* No GPU training or dataset preparation required.
- *Flexibility:* Vocabulary can be updated per recording without model redeployment.
- *Targeted improvement:* Reduces substitution errors specifically for low-frequency domain terms without affecting general vocabulary.

We quantify the effectiveness of this approach in Section 5.4. Illustrative English ASR and enrichment prompts are given in Appendix B.

3.4 Content Enrichment: Summarization and Open-ended Questions

For extracting thematic structure from transcripts, we employ the DeepSeek V3 large language model (DeepSeek, 2024). The model receives the full transcript with timestamps and generates a structured list of topics with start and end time markers for each thematic block. Processing cost for

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

topic extraction averages about $ 0.019 per hour of content (converted from operational accounting at 79 RUB/USD).

Beyond topic segmentation, we leverage the same model to produce two additional pedagogical artifacts with minimal additional token consumption: concise lecture summaries (200-300 words) and 3-4 open-ended questions with answer keys. Both are generated in a single model pass. Summaries help students review content efficiently, while open-ended questions enable deeper comprehension checks – functionality that transforms raw transcripts into structured learning materials. Total cost for all enrichment tasks (topic extraction, summarization, question generation) averages about $ 0.019 per hour of content, bringing the combined cost for transcription and enrichment to about $ 0.075 per hour (transcription about $ 0.056/h plus enrichment about $ 0.019/h at 79 RUB/USD). Based on transcription results, subtitle files in SRT and VTT formats with sentence-level segmentation are automatically generated.

4 Platform Architecture

4.1 System Design

The platform is implemented using: FastAPI (Ramírez, 2026) for asynchronous REST API; SQLAlchemy 2.0 for asynchronous ORM operations; PostgreSQL (PostgreSQL, 2026) with JSONB for flexible metadata storage; Redis (Redis, 2026) as message broker; and Celery (Celery, 2026) for distributed task processing. The architecture follows the multi-layer pattern (Fowler, 2002): API layer, service layer, and repository layer with automatic user ID filtering for multi-tenancy. Transcription and enrichment are invoked through provider-specific adapters so that alternative ASR or LLM endpoints can be selected per template or deployment without changing the orchestration graph.

Configuration management employs a hierarchical resolver that merges settings from three levels: user defaults, processing templates, and per-recording overrides. This enables controlled experimentation with different preprocessing parameters and model configurations while maintaining reproducibility. Feature flags allow gradual rollout of new capabilities and A/B testing of processing strategies across user cohorts.

Data isolation is enforced at three levels: database (automatic filtering by user_id), service layer (user context propagation), and file system (separate directory structures per tenant). Soft delete with configurable retention periods ensures data recoverability, with hard deletion performed by periodic maintenance tasks.

4.2 Parallel Pipeline Processing

To maximize processing speed, we implemented the Celery Chains pattern (Celery, 2026). Rather than a monolithic task blocking worker processes, processing is decomposed into atomic stages as shown in the simplified schema in Figure 1.

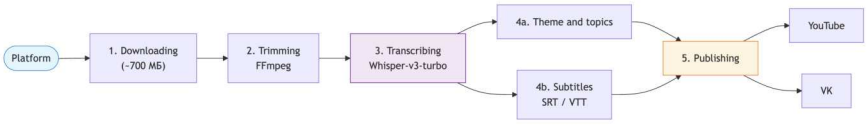


Figure 1: Simplified processing pipeline: download → video trimming → transcription → parallel group (topic extraction + subtitle generation + content enrichment) → publication

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

The orchestrator constructs the chain in approximately 0.08 seconds and immediately releases the worker process. Topic extraction, subtitle generation, and content enrichment execute in parallel, as all depend on transcription results but are mutually independent.

The platform's ability to sustain concurrent processing of up to 50 recordings stems from its event-driven, asynchronous architecture with dedicated worker pools for different task types. We decompose the pipeline into discrete, atomic tasks orchestrated via Celery Chains, which yields two critical properties:

- *Non-blocking orchestration:* The main process launches the entire chain in sub-second time and is immediately freed, preventing request queuing.
- *Elastic resource allocation:* CPU-bound tasks (FFmpeg trimming, audio preprocessing) are dispatched to a dedicated pool of 6 prefork workers (celery-cpu), while I/O-bound tasks (API calls to Fireworks, DeepSeek; downloads; uploads) are handled asynchronously by separate thread-based pools (20 threads for downloads (celery-downloads), 20 for uploads (celery-uploads), and 28 for API calls (celery-async)). This separation ensures that a spike in video uploads does not starve the transcription workers, and vice-versa.

Under our current configuration, the system achieves linear throughput scaling up to 50 concurrent recordings, with end-to-end latency remaining under 2 minutes per recording. Load testing reveals that the CPU-bound trimming stage becomes the bottleneck beyond this threshold: at 60–70 concurrent recordings, queue wait times increase proportionally, but the system remains stable with no task failures or timeouts. This graceful degradation ensures reliable operation even under unexpected load spikes.

The architecture supports horizontal scaling to address higher sustained loads. By increasing the number of CPU worker nodes or deploying more powerful CPU-optimized instances, the platform can linearly expand its capacity. This design ensures that the system can accommodate growing institutional demand without requiring changes to the core pipeline logic.

4.3 Reliability and Error Handling

The pipeline implements stage-specific failure strategies to ensure robustness:

- *Download failures:* Recording marked as INITIALIZED, preserving upstream state for manual retry.
- *Trim failures:* State reverted to DOWNLOADED, avoiding redundant downloads.
- *Transcription failures:* With configurable `allow_errors` flag, system can skip dependent stages (topic extraction, subtitle generation) rather than failing the entire pipeline.
- *Upload failures:* Per-target FAILED state allows selective retries without re-processing.

A centralized failure reset mechanism determines which stages require re-execution based on error type and stage dependencies. All processing stages are instrumented with an append-only timing audit log recording stage type, substep, attempt number, and duration in milliseconds. This instrumentation provides the empirical data for the performance measurements reported in Section 5.5.

5 Experimental Methodology and Results

5.1 Dataset Description

Experiments were conducted on a corpus of over 700 real educational video recordings (totaling more than 900 hours of content). The dataset exhibits substantial diversity:

- *Duration.* Recordings range from 1 to 3 hours, with mean duration of 94 minutes (approximately 1.57 hours).
- *Subject coverage.* Content spans humanities, social sciences, and technical disciplines, featuring domain-specific terminology in each field.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

- *Audio quality.* The majority of recordings (61%) feature good audio quality (e.g., studio or high-quality headset microphones). A further 23% are classroom recordings with moderate background noise, and the remaining 16% are very quiet or have significant background interference (e.g., poor remote conferencing setups).
- *Speaker diversity.* Approximately 70 unique speakers across recordings.

5.2 Experimental Setup

Recordings were processed in the platform's operational mode with parallel batch loading (up to 50 recordings simultaneously). For each recording, we measured execution time per pipeline stage using the built-in timing audit system, which records duration with millisecond precision. All reported metrics (processing time, speedup factor, API costs, labor savings) are averaged over the full corpus of over 700 recordings. Competitor cost comparisons are based on published tariffs as of March 2026.

Measurement hardware. Primary timing used a cloud Linux VM (Ubuntu 24.04 LTS, AMD x86_64 8 vCPU, 8 GB RAM, 100 GB SSD) with stable connectivity to object storage and APIs. The same Celery topology applied everywhere: six `prefork` workers for FFmpeg trimming and preprocessing, with separate thread pools for downloads, uploads, and asynchronous API calls (Section 4). A laptop on a consumer link showed large variance in download and upload stages. Under quiet network, laptop versus VM runs matched within ± 6 s for CPU-bound orchestration and trimming on the same recording. Remote ASR and LLM run on provider infrastructure, so local hardware affects I/O and trimming, not acoustic RTF. Table 3 averages over the full corpus on the VM. Batch stress tests on the VM (on the order of ten to twenty concurrent recordings) showed stable throughput and success rates, consistent with engineering logs.

5.3 Sensitivity Analysis

We evaluated the impact of preprocessing parameters on transcription quality and cost:

- *Silence removal threshold.* Using a threshold of -32 dB, we achieved an average duration reduction of 15.53% across the corpus, with less than 0.5% Word Error Rate (WER) impact. Lecture recordings frequently contain extended periods of silence – waiting for an audience to settle, long pauses between topics, or delays before stopping – which our pipeline automatically removes. More aggressive settings caused speech truncation.
- *Compression ratio.* Dynamic range compression with 2:1 ratio improved recognition accuracy on low-quality recordings by 4.7% relative WER reduction, while excessive compression (4:1) introduced artifacts that increased WER by 2.1%.
- *Noise reduction method.* Spectral gating reduced background noise by 8.2 dB with minimal speech distortion, incurring only 1.1% WER increase. This confirms the "noise reduction paradox" – aggressive cleaning can harm ASR performance despite improving objective SNR metrics.
- *Audio format selection.* Transcoding to 16 kHz mono MP3 at 64 kbps (`libmp3lame`), i.e. the same compressed stream uploaded to the ASR API after extraction (Section 3.2), achieved 95.8% size reduction compared to uncompressed WAV at 48 kHz stereo, with negligible quality impact (less than 0.3% WER difference). This dramatic compression enables faster uploads and reduces storage costs while preserving transcription accuracy.

Empirical Result 1 (Preprocessing cost–quality trade-off) *Audio preprocessing (silence removal at -32 dB, dynamic range compression 2:1) yields 10–25% reduction in billable duration with negligible accuracy loss (WER increase $< 0.5\%$). Transcoding to 16 kHz mono MP3 at 64 kbps preserves accuracy while enabling $> 95\%$ size reduction.*

Observation 1 (Noise reduction paradox) *Spectral gating reduces background noise by 8.2 dB but incurs 1.1% WER increase. Aggressive cleaning can harm ASR despite improving SNR.*

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

5.4 Impact of Video Vocabulary

To quantify the benefit of our prompt-based domain adaptation, we drew 20 lectures at random from different courses among those with dense technical terminology (computer science, engineering, mathematics, and biology). Instructors supplied course-specific terms (specialty vocabulary, not only generic syllabus headings). We formed a Video Vocabulary of 5-10 key terms per lecture from that input. Processing these lectures with the vocabulary prompt resulted in:

- 15.7% relative reduction in terminology-related errors, as measured by a specialized Character Error Rate (CER) on the keyword set.
- 3.2% absolute WER reduction on the full transcripts of technical lectures.
- Negligible impact on lectures without specialized terminology, confirming that prompting does not degrade general-domain accuracy.

Empirical Result 2 (Video Vocabulary effectiveness) *Prompt-based injection of* 5–10 *key terms (whole-request scope, no fine-tuning) yields* 15.7% *relative CER reduction on keywords and* 3.2% *absolute WER reduction on technical lectures. On lectures without specialized terminology, the impact is negligible.*

5.5 Performance Results

Table 3: Experimental evaluation results (per hour of content)

Metric	Manual baseline	Our pipeline
Processing time per hour	30 min	1 min 22 sec*
Parallel processing capacity	1 recording	up to 50 recordings
End-to-end speedup	—	$\sim 22\times$
API cost per hour (transcription + enrichment)	—	$\sim$\$0.075**
Manual post-production labor (100 recordings / month)	50+ person-hours	automated
Verification / spot-check time (same volume)	—	up to 1 hour***

*Per hour of lecture content, mean wall-clock 82 s (22 download, 10 trim, 12 transcribe, 12 enrich, 26 publish). Speedup vs. the manual-time row uses the same nominal hour of content: $(30 \times 60\,\mathrm{s})/(82\,\mathrm{s}) \approx 22$.
**About \$ 0.075/h total: $\sim$\$ 0.056 transcription + $\sim$\$ 0.019 LLM enrichment; RUB accounting converted at 79 per USD.
***Transcription and enrichment run without manual steps; for the same monthly batch scale as the manual row, staff spend at most $\sim$1 h/month on spot-checks versus 50+ person-hours for fully manual post-production.

Empirical Result 3 (End-to-end pipeline performance) *Our measurements show that on a corpus of* $700+$ *recordings (*$900+$ *hours), the pipeline reduces post-production cycle from* 30 *min to under* 2 *min per recording, achieves* $\sim\ 22\times$ *speedup, costs* $\approx\ \$\,0.075$*/h (transcription + topic extraction + summary + self-check questions), supports up to* 50 *concurrent recordings, and is deployed in production.*

5.6 User Feedback Study

To gather exploratory evidence of practical utility after deployment, we distributed a voluntary post-deployment questionnaire (not a randomized controlled trial) to students and teaching assistants who had access to the platform's outputs (chapter topics, timestamps, summaries and self-check questions). Respondents rated several aspects on Likert scales, including overall satisfaction with the bundle of artifacts and perceived accuracy of topics and timestamps.

These results are descriptive post-deployment feedback only; limitations on design, instruments, and bias are summarized in Section 6 below. They nonetheless indicate broad acceptance of the generated materials in our deployment context.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Figure 1. Distribution of User Satisfaction

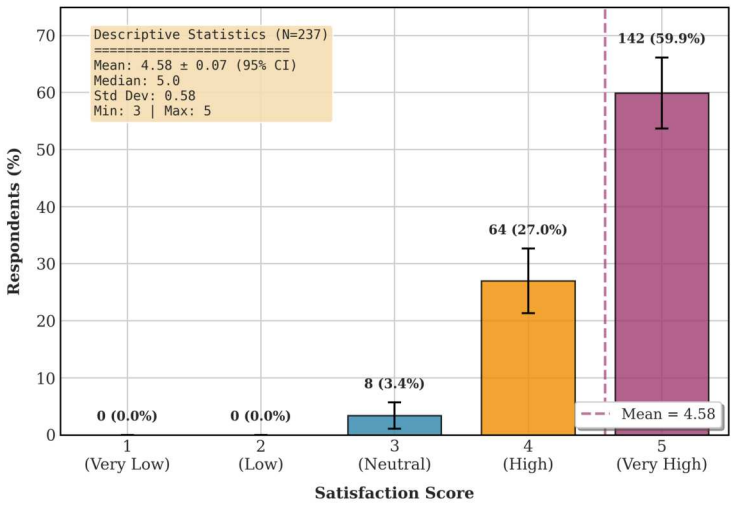


Figure 2: Overall satisfaction with the bundle "topics + timestamps + summary + questions" (n=237): mean 4.58 (95% CI $[4.502, 4.650]$), median 5.0, SD 0.58; 59.9% rated 5 (Very High), 27.0% rated 4, 3.4% rated 3; none rated 1-2.

6 Limitations

While our technical evaluation demonstrates robust performance and the user feedback indicates positive reception, several limitations should be noted:

- *Language specificity.* Evaluation focused primarily on Russian-language content, though over 50 hours of English lectures have also been processed. Performance on other languages requires further investigation.
- *Terminology handling.* While our Video Vocabulary mechanism significantly reduces terminology errors, highly specialized or out-of-vocabulary terms occasionally suffer recognition errors. As demonstrated by Tobias (2025), domain-specific fine-tuning can reduce such errors further (from 5.78% to 4.53% WER), suggesting a complementary direction for future work.
- *Speaker dependency.* Non-native speakers and strong regional accents show 5–8% higher WER in our internal spot checks. Broader ASR literature documents systematic performance gaps on accented and non-native speech relative to mainstream reference varieties (Hinsvark et al., 2021); educational lecture recordings inherit the same structural risk (Rao, 2023). Mitigation (human review, glossary prompts, or targeted adaptation) remains future work.
- *Computational requirements.* Peak loads exceeding 50 simultaneous recordings require dynamic scaling mechanisms not yet implemented, though the architecture supports horizontal scaling to address this.
- *User feedback scope.* The questionnaire was administered after deployment; it is not an RCT, did not use a validated instrument such as SUS, and may suffer from self-selection and social-desirability bias. We report it only as early descriptive evidence of acceptance, not as proof of pedagogical impact.
- *Cloud API vs. acoustic model.* Managed ASR endpoints couple proprietary preprocessing, segmentation, and serving versions with an advertised backend. Our benchmark emphasizes tariffs, latency documentation, and features; it does not isolate neural weights from provider infrastructure without a separate controlled study.
- *Data privacy and ethics.* Voice signals can be sensitive: in addition to linguistic content, they may support speaker identification if leaked, which is why voice is often treated as

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

a biometric modality in privacy regulation. Our platform stores raw audio on our infrastructure for a configurable retention window (default 30 days); deployers may shorten or extend this policy. Users are notified via terms of service. Audio is transmitted to third-party ASR/LLM APIs only for processing; providers are contractually prohibited from using customer data for model training. All students in the programs where the platform operates have signed consent agreements for voice recording as part of their online program enrollment; future deployments may add explicit opt-in for each new processing service. We do not present a quantitative re-identification risk analysis; this is left to institutional risk review.

7 Conclusion

This paper compared commercial speech recognition APIs for educational lecture video (Table 1) and described a production pipeline for transcription, subtitles, and pedagogical enrichment. We motivated managed Whisper-style endpoints under our tariff and multilingual constraints, evaluated audio preprocessing and prompt-based Video Vocabulary on a corpus of over 700 recordings (Sections 5-5.4), and reported end-to-end speedup and cost relative to manual post-production (Empirical Result 3, Table 3). The Fireworks-managed `whisper-v3-turbo` endpoint was the operational choice for this deployment; Section 3 explains how that choice relates to published prices, latency documentation, and prompting features.

The resulting system is deliberately flexible. The architecture follows a multi-layer service pattern (Fowler, 2002) with provider-specific adapters for ASR and LLM calls (Section 4), so the orchestration graph stays stable when backends change. Configuration can already select different templates and endpoints per deployment or course. Future work will use that surface to route languages or programs to different ASR engines or cost-quality tiers where needed, and to combine managed APIs with on-premise or open-weight engines when privacy, tariffs, or accuracy trade-offs warrant it – without a ground-up redesign of the pipeline.

Exploratory post-deployment ratings (n=237, Figure 2) suggest broad acceptance of the generated materials; they are not evidence of learning gains. The platform runs in production on real lecture content. *Future work* includes controlled studies of learning outcomes; extending Video Vocabulary to additional languages and domains; auto-scaling worker pools from queue depth; complementary fine-tuning for extreme terminology; and empirical evaluation of multi-backend routing policies under load.

A RTF Measurement Methodology for Self-Hosted Systems

Real-time factor (RTF) is defined as the ratio of processing time to audio duration: RTF $= t_{\text{process}}/t_{\text{audio}}$. For self-hosted systems, RTF is highly dependent on hardware configuration. Table 4 summarizes documented benchmarks from the literature and official repositories.

Table 4: Documented RTF benchmarks for self-hosted ASR systems

Model	Hardware	Conditions	RTF
Faster-Whisper large-v2	NVIDIA RTX 3070 Ti 8GB	13 min audio, fp16, beam=5	0.081 (SYSTRAN, 2026)
Faster-Whisper large-v2	NVIDIA RTX 3070 Ti 8GB	13 min, fp16, batch=8	0.022 (SYSTRAN, 2026)
Faster-Whisper large-v2	NVIDIA RTX 3070 Ti 8GB	13 min, int8, batch=8	0.021 (SYSTRAN, 2026)
Faster-Whisper small	Intel i7-12700K, 8 threads	13 min, int8, batch=8	∼0.14 (SYSTRAN, 2026)
WhisperX large-v2	GPU <8GB VRAM	70× realtime with batch	∼0.014 (Bain et al., 2023)
Vosk	CPU (Ryzen 5 5600G)	3 min audio	0.3–0.5 (Vosk, 2020)
GigaAM (encoder only)	CUDA GPU	Encoder: 10 ms / 10 s (bs=1)	0.001 (single) (Salute Developers, 2025)
Pisets	GPU (type not specified)	Depends on GPU	0.15–0.25 (Bondarenko et al., 2025)
Pisets	CPU	Depends on CPU	1.0–1.5 (Bondarenko et al., 2025)

All RTF values are reported as $\text{time}_{\text{process}}/\text{time}_{\text{audio}}$. For API services, RTF is estimated from provider latency data (e.g., Fireworks reports 3–7 seconds for 1 hour of audio, RTF = 0.00083–0.002) (Fireworks AI, 2026).

These benchmarks demonstrate that RTF varies by orders of magnitude depending on hardware (GPU vs. CPU), quantization (fp16 vs. int8), batching, and model size. For production deployment

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

at scale, self-hosted stacks require careful hardware provisioning; cloud APIs offer predictable per-minute pricing and elastic scaling without infrastructure management.

B Illustrative Prompts (English Lectures)

Production prompts are written in the lecture language. For English audio and downstream enrichment, we use English strings; for Russian lectures, equivalent Russian templates apply. Tables 5 and 6 show synthetic examples (no real course or person names).

B.1 Transcription ASR `prompt` (English)

Video Vocabulary lists 5-10 comma-separated terms from the syllabus and is passed as the `prompt` field to the Fireworks transcription endpoint (Fireworks AI, 2026), with whole-request scope as in Section 3. When the deployer does not supply a custom base string, we prepend a short English course-context line (title and spelling guidance), then the vocabulary list. Table 5 shows synthetic fragments for English-language transcription; the final `prompt` concatenates these parts.

Table 5: Synthetic ASR `prompt` fragments for English transcription (concatenated in production).

Base context	`This is a video lecture from the course ''Machine Learning 101''. Preserve correct spelling of domain terms (including proper nouns), abbreviations and technical terms. Account for features of spoken language: incomplete sentences, pauses, natural intonation.`
Vocabulary	`gradient descent, loss function, overfitting, regularization, PyTorch, expectation, variance`

B.2 Chapter topics and enrichment (English)

Topic timestamps, summaries, and self-check questions are produced by an LLM (DeepSeek V3 in production (DeepSeek, 2024)) from the transcript. The user message follows a fixed outline: summary, a single main theme title, 5-8 timestamped chapter lines `[HH:MM:SS] - Title` using only timestamps present in the transcript, then numbered open-ended questions. Table 6 abbreviates the instruction block; the full template fills duration and count parameters from processing settings.

Table 6: Synthetic instruction skeleton for topic timestamps and enrichment (English lectures).

```
Analyze the English lecture transcript below. Output: (1) a 2-4
sentence summary; (2) one main theme title (2-5 words); (3) 5-8
chapter topics as lines ''[HH:MM:SS] - Title'', using only
timestamps that appear in the transcript, in chronological order;
(4) 3-4 open-ended self-check questions. Then paste the
transcript.
[TRANSCRIPT]
```

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

REFERENCES

S. Dhawan. Online Learning: A Panacea in the Time of COVID-19 Crisis. *Journal of Educational Technology Systems*, 49(1):5–22, 2020.

H. Nagataki, T. Nagai, and N. Tokura. An Effort for Recording and Delivering of Lectures with Student Staff at Tottori University of Environmental Studies. *Japan Journal of Educational Technology*, 2006.

A. Holmberg. Generating Narrated Lecture Videos from Slides with Synchronized Highlights. *arXiv preprint arXiv:2505.02966*, 2025.

Z. M. Tobias. Domain-Specific Customization for Improving Speech to Text. Master's thesis, Rochester Institute of Technology, 2025.

A. Rao. Transcribing Educational Videos Using Whisper: A preliminary study on using AI for transcribing educational videos. *arXiv preprint arXiv:2307.03200*, 2023.

A. Hinsvark et al. Accented Speech Recognition: A Survey. *arXiv preprint arXiv:2104.10747*, 2021.

M. Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley, Boston, 2002.

Fireworks AI. Audio Transcription Launch. URL: `https://fireworks.ai/blog/audio-transcription-launch`, 2026.

OpenAI. Audio API Documentation. URL: `https://platform.openai.com/docs/api-reference/audio`, 2026.

Google Cloud. Speech-to-Text Documentation. URL: `https://cloud.google.com/speech-to-text/docs`, 2026.

AssemblyAI. Universal-3 Pro Documentation. URL: `https://www.assemblyai.com/docs/`, 2026.

Gladia. Documentation. URL: `https://docs.gladia.io/`, 2026.

Deepgram. Nova-3 Documentation. URL: `https://developers.deepgram.com/docs/`, 2026.

ElevenLabs. Speech-to-Text Capabilities. URL: `https://elevenlabs.io/docs/overview/capabilities/speech-to-text`, 2026.

A. Radford, J. W. Kim, T. Xu, et al. Robust Speech Recognition via Large-Scale Weak Supervision. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 28492–28518, 2023.

Vosk. Offline Speech Recognition Toolkit. URL: `https://alphacephei.com/vosk/`, 2026.

Salute Developers. GigaAM-v3: Russian ASR Model. URL: `https://developers.sber.ru/kak-v-sbere/culture/gigaAM-v3`, 2025.

T-Bank. T-one: Streaming Russian ASR. URL: `https://github.com/voicekit-team/T-one`, 2025.

I. Bondarenko. whisper-podlodka-turbo. Hugging Face, URL: `https://huggingface.co/bond005/whisper-podlodka-turbo`, 2025.

I. Kuleshov and A. Nikolich. Borealis: Audio LLM for Russian ASR. Hugging Face, URL: `https://huggingface.co/Vikhrmodels/Borealis`, 2025.

M. Bain, J. Huh, T. Han, and A. Zisserman. WhisperX: Time-Accurate Speech Transcription of Long-Form Audio. In *Proceedings of Interspeech*, 2023.

SYSTRAN. Faster-Whisper: CTranslate2 Implementation. URL: `https://github.com/SYSTRAN/faster-whisper`, 2026.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1006 DOI: 10.66693/mathai.1006
Special Issue: MathAI 2026 Selected Papers
Received: 02 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

SYSTRAN. Faster-Whisper README. URL: https://github.com/SYSTRAN/faster-whisper, 2026.

I. Bondarenko, D. Grebenkin, O. Sedukhin, M. Klementev, R. Derunets, and L. Budneva. Pisets: A Robust ASR System for Lectures and Interviews. In *Proceedings of NAACL (Industry Track)*, 2025.

Fireworks AI. Pricing. URL: https://fireworks.ai/pricing, 2026.

OpenAI. API Pricing. URL: https://openai.com/api/pricing/, 2026.

AssemblyAI. Pricing. URL: https://www.assemblyai.com/pricing, 2026.

AssemblyAI. Universal-3 Pro Prompting. URL: https://www.assemblyai.com/docs/streaming/universal-3-pro/prompting, 2026.

Deepgram. Pricing. URL: https://deepgram.com/pricing, 2026.

TalkFlow AI. Deepgram Nova-3 Review: Benchmarks and Pricing. URL: https://transcriber.talkflowai.com/blog/deepgram-nova-3-review-benchmarks-pricing, 2025.

Deepgram. Keyterm Prompting. URL: https://developers.deepgram.com/docs/keyterm, 2026.

Cloudflare. Nova-3 Workers AI Model. URL: https://developers.cloudflare.com/workers-ai/models/nova-3/, 2026.

Gladia. Pricing. URL: https://www.gladia.io/pricing, 2026.

ElevenLabs. API Pricing. URL: https://elevenlabs.io/pricing/api, 2026.

Google Cloud. Speech Adaptation. URL: https://cloud.google.com/speech-to-text/v2/docs/adaptation-model, 2026.

Salute Developers. GigaAM Evaluation. URL: https://github.com/salute-developers/GigaAM/blob/main/evaluation.md, 2025.

Vosk. Issue #1007: RTF on CPU. URL: https://github.com/alphacep/vosk-api/issues/1007, 2020.

Alphacephei. Russian ASR Models Benchmark (11 Domains). URL: https://alphacephei.com/nsh/2025/04/18/russian-models.html, 2025.

I. Bondarenko et al. Pisets: source repository (performance and usage). URL: https://github.com/bond005/pisets, 2025.

S. Gandhi, P. von Platen, and A. M. Rush. Distil-Whisper: Robust Knowledge Distillation via Large-Scale Pseudo Labelling. *arXiv preprint arXiv:2311.00430*, 2023.

FFmpeg: A complete, cross-platform solution to record, convert and stream audio and video. URL: https://ffmpeg.org/, 2026.

Fireworks AI. Audio Transcriptions API Reference. URL: https://docs.fireworks.ai/api-reference/audio-transcriptions, 2026.

DeepSeek. DeepSeek-V3 Technical Report. *arXiv preprint arXiv:2412.19437*, 2024.

S. Ramírez. FastAPI Documentation. URL: https://fastapi.tiangolo.com/, 2026.

The PostgreSQL Global Development Group. PostgreSQL Documentation. URL: https://www.postgresql.org/, 2026.

Redis Ltd. Redis Documentation. URL: https://redis.io/, 2026.

Celery: Distributed Task Queue. URL: https://docs.celeryq.dev/, 2026.

LEAP – Lecture Enhancement & Automation Platform. Source code and documentation. https://github.com/GordeyZuev/LEAP, 2026.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007

Special Issue: MathAI 2026 Selected Papers

Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

The Evolution of Mind: Emergence of Collective Intelligence through Logical-Probabilistic Knowledge Dynamics in Multi-Agent ENIGMA Metaverse Ecosystems

Andrey Nechesov[1], Sergey Barykin[1] and Janne Ruponen[1]

[1] *International Artificial Intelligence Committee (IAIC)*

nechesoff@gmail.com, sbe@list.ru, ruponez@gmail.com

Abstract

Modern metaverse platforms, populated by heterogeneous multi-agent systems (MAS), generate vast streams of experiential data whose epistemic value remains largely untapped. This paper introduces the **Enigma framework**—a formal theory of *collective intelligence emergence* in metaverse ecosystems, grounded in a novel **logical-probabilistic learning theory** that extends classical first-order logic with probabilistic confidence annotations and distributed knowledge semantics. We define a *Distributed Knowledge Lattice* (DKL) over multi-agent interactions [33] and prove that, under precisely stated monotonicity and convergence conditions, the collective knowledge of an agent population forms a complete lattice whose least upper bound represents an emergent cognitive state unreachable by any individual agent. We formalize the dynamics of knowledge creation, verification, and propagation through a system of *Logical-Probabilistic Agents* (LP-agents) that interact with LLM-driven entities, providing trustworthy and explainable reasoning via symbolic proof chains stored on a multi-blockchain ledger. Central results include: (i) a **Collective Intelligence Convergence Theorem** establishing conditions under which the system's aggregate knowledge monotonically approaches a fixed point; (ii) a **Probabilistic Inference Soundness Theorem** guaranteeing that confidence propagation through distributed reasoning chains preserves logical consistency; and (iii) a polynomial-time algorithm for optimal knowledge retrieval from the distributed lattice. The framework is instantiated within the Enigma Metaverse architecture, where smart contracts govern knowledge tokenization, cross-chain knowledge interoperability protocols enable seamless sharing, and decentralized governance mechanisms ensure epistemic accountability. We demonstrate that this synthesis of mathematical logic, probability theory, LLM-based hypothesis generation, and blockchain-secured knowledge persistence provides a rigorous foundation for building self-optimizing, trustworthy, and explainable collective intelligence (CI) in virtual worlds.

Keywords: Collective Intelligence, Metaverse, Multi-Agent Systems, Logical-Probabilistic Learning Theory, Distributed Knowledge Lattice, Explainable AI, Trustworthy AI, Large Language Models, Blockchain, Smart Contracts

1 Introduction

The pursuit of artificial intelligence systems that exhibit genuine collective cognition—intelligence that transcends the sum of its individual agents—represents one of the deepest open problems at the intersection of mathematical logic, probability theory, and distributed computing [16, 21]. While contemporary Large Language Models (LLMs) demonstrate impressive individual reasoning capabilities [1, 20], they remain fundamentally isolated: each model instance operates within its own ephemeral context window, lacking mechanisms for cumulative, verifiable knowledge sharing with other intelligent entities.

The metaverse—conceived as a persistent, immersive virtual universe populated by autonomous agents [24]—offers a unique substrate for studying and engineering collective intelligence [10, 12]. Unlike static benchmarks, a metaverse environment demands that agents continuously learn, adapt, cooperate, and compete within a shared world governed by formal rules. This paper argues that the metaverse is not merely a testbed for AI but an *active cognitive ecosystem* where collective intelligence (CI) naturally emerges from the structured interactions of heterogeneous agents.

Received: March 15, 2026 Accepted: April 1, 2026 Published: April 4, 2026

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

The central challenge we address is: *Under what mathematical conditions does a population of interacting agents, each possessing partial and probabilistic knowledge, give rise to a coherent CI that is provably more capable than any individual?* Answering this requires a formal framework that unifies three traditionally separate domains:

(i) **Mathematical Logic**: providing the inferential backbone for rigorous, verifiable reasoning;

(ii) **Probability Theory**: accounting for the inherent uncertainty in agent-generated knowledge;

(iii) **Distributed Systems**: ensuring that collective knowledge is persistent, tamper-proof, and accessible.

Our approach builds upon and substantially extends the task-based cognitive architecture and the logical-probabilistic learning theory introduced for virtual cities [12], while proposing a fundamentally new mathematical object—the **Distributed Knowledge Lattice** (DKL)—as the formal substrate for CI.

1.1 Contributions

The principal contributions of this paper are:

1. **A Formal Theory of CI Emergence**: We define CI as a fixed-point property of a knowledge lattice and prove convergence under realistic conditions (Section 3).
2. **Logical-Probabilistic Knowledge Dynamics**: We introduce a dynamical system governing the evolution of probabilistic knowledge across a multi-agent population, with soundness guarantees for distributed inference (Section 4).
3. **LP-Agent Verification Framework**: We formalize the interaction between LLM-agents and Logical-Probabilistic agents, providing explainability [26] and trustworthiness [25] guarantees (Section 5).
4. **The Enigma Architecture**: We instantiate the theory within a concrete multi-blockchain metaverse platform with smart-contract-governed knowledge economics (Section 6).
5. **Convergence Analysis and Complexity Results**: We provide precise convergence rates and prove that optimal knowledge retrieval from the DKL is polynomial (Section 7).

2 Related Work

Neuro-Symbolic AI and LLM Verification. The integration of neural generation with symbolic verification has gained significant traction. Logical Neural Networks [15, 23] embed propositional logic directly into network architectures. Logic-LM [13] and PAL [4] use external symbolic solvers to verify LLM outputs. The TBCA framework established a closed cognitive loop where LLMs generate hypotheses verified by a symbolic engine, with results stored on blockchain. Our work extends this by formalizing the *inter-agent* dynamics of knowledge creation and proving emergent properties of the collective.

Multi-Agent Systems and CI. Classical MAS theory [17, 21] addresses coordination, negotiation, and emergent behavior but typically lacks formal treatments of knowledge dynamics. Recent work on generative agents [32] and CAMEL [11] demonstrates emergent social behaviors in LLM-based agent societies. Project Sid [22] simulates civilizations with 1000 agents. However, these works lack formal guarantees on knowledge quality, convergence, or trustworthiness.

Blockchain-Based Knowledge Management. The axiomatization of blockchain theory [6] provides formal foundations for distributed ledger properties. Applications to AI knowledge management [9] demonstrate tamper-proof knowledge sharing. Our work introduces *knowledge tokenization* as an economic mechanism for incentivizing CI growth.

Metaverse as AI Research Platform. Virtual cities [12] and digital twins [7, 18] provide immersive environments for AI testing. The concept of Artificial Collective Consciousness (ACC) [12] and DAO-based governance [2] inform our approach to epistemic governance in the Enigma ecosystem.

Probabilistic Logic and Learning Theory. Probabilistic logic programming [3] and statistical relational learning [5] combine logical structure with probabilistic inference. Our logical-probabilistic learning theory draws from the task-based approach [12] but extends it with lattice-theoretic semantics and distributed consensus properties.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

The Openness Problem. Multi-agent systems in open environments face three types of dynamism: 1) Agents join (onboarding) and leave continuously (exit and slashing), violating the fixed set assumption of traditional MARL; 2) Transactions appear (submission) and disappear (replacement, reorganization) dynamically, breaking the stable reward function assumption; and 3) evolution of validator capability (upgrades), preference shift (MEV strategies), and goal adjustment (honest to Byzantine) invalidating mapping the coherent action-outcome dynamics. Studies show that openness degrades MARL performance by 40% for single openness types, while catastrophic collapse under combined openness [27]. This quantifies the cost of decentralization and motivates architectural solutions.

Credit Assignment Problem. Credit assignment attributes shared rewards to individual agent contributions, and is identified as the multi-agent coordination problem [28, 29, 31]. This problem is divided into temporal credit assignment (TCA) and structural credit assignment (SCA) [27]. TCA addresses delayed rewards by attributing a reward received at time t to actions taken a priori $t - k$. In blockchain, TCA links validator actions to future block rewards despite epoch delays. SCA addresses multi-agent attribution by distributing a shared team reward among individual contributions by agents.

Traditional approaches (VDN, QMIX) fail under openness due of assumption of stable team structure [27]. Polarization policy gradients (MAPPG) solve this by amplifying the reward gap between optimal and non-optimal behaviors [29] via memory substrates that can track agent contributions over time.

Persistent Memory in Multi-Agent Coordination Recent research demonstrate that cognitive CI in MAS emerges from persistent agent interactions when equipped with proper memory architectures [32, 35]. However, typical MAS implementations reset memory after each session, preventing the formation of long-term relationships, knowledge accumulation, and adaptive coordination strategies that define intelligent systems.

While MADRL focuses on training paradigms and reward-driven behaviors, LLM-based MAS adds distinct challenges centered on memory management and contextual reasoning. Memory types have been divided in earlier studies to five categories: short-term memory for live interactions, long-term memory storing historical queries, external data storage for knowledge augmentation, episodic memory capturing past events of multi-agent interactions, and consensus memory providing distributed knowledge across agents [30]. These memory types operate within complex hierarchical access control, and are required to maintain consistency across overlapping agent knowledge while enabling effective retrieval of past interactions based on contextual relevance.

Memory is essential for agents to recall prior subtasks, coordinate handoffs, maintain role histories, and build on earlier steps rather than starting from scratch [30], yet virtually all current MAS implementations treat memory in session-limited scope, in which agents reset each interaction. There is no architecture in the literature providing a hardware-efficient, distributed, always-on memory substrate at the agent level. Management of memory storage has been identified as playing a critical role in collaboration between agents, providing constraints to context alignment, adaptive learning and associative recall [30].

Attempts to establish persistent agent memory in simulated social environments have relied on recency, importance, and reflection as retrieval signals [32]. However, this strategy increases computational costs as each memory query requires scoring the entire history with complex weighting functions, while no providing mechanisms for forgetting obsolete information or representing uncertainty.

Game theory combined with blockchain networks constitutes partially observable stochastic games in which validators must coordinate under uncertainty [31]. The non-stationarity inherent in open blockchain systems violates the assumptions of traditional reinforcement learning [27], requesting novel memory architectures for dynamic, unbounded agent populations.

MARL Problems. Multi-agent reinforcement learning (MARL) surveys have identified existing challenges stemming from credit assignment problem (CAP), non-stationary interactions, and scalability due to exponentially growing state-action space [34]. MARL has undergone a rapid transformation since the introduction of deep learning (DL) methods, enabling agents to act on real-world complexity that was previously unmanageable with traditional tabular approaches. In multi-agent deep reinforcement learning (MADRL), autonomous agents interact simultaneously within a shared environment while learning policies through trial-and-error and adapting to the behavioral changes of other agents. The one of the applicable frameworks for MADRL is the Markov Game, which extends the single-agent Markov Decision Process (MDP) to multiple decision-makers, each with individual action spaces and reward functions [34].

MADRL depends on the training paradigm employed to learn agent policies. Gronauer and Diepold identify three primary approaches: distributed training with decentralized execution (DTDE), where agents learn independently;

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

centralized training with centralized execution (CTCE), where a single joint policy governs all agents; and centralized training with decentralized execution (CTDE), which has emerged as the state-of-the-art approach [34]. The CTDE paradigm allows agents to share information such as events, functions and policy parameters during training while acting independently at deployment. This addresses the challenge of non-stationarity, where the environment appears to change from each agent's perspective as others simultaneously adjust their own policies.

The CTDE paradigm resolves the centralized-decentralized mismatch problem, where weak agent policies corrupt reward attribution for others. However, its applicability is limited in open blockchain environments where validator sets rotate continuously as CTDE assumes stable team structure during training.

Non-Stationarity and Experience Replay Obsolescence The non-stationarity problem in MARL emerge as each agent's environment includes the policies of other agents, each evolving individually during learning [31]. The state transition function from agent's perspective becomes non-stationary. Meanwhile, typical experience replay stores transitions and samples them uniformly during training. However, when contrasting policies evolve, stored experiences return outdated strategies. For example, a validator's learned response to a transaction submission pattern becomes obsolete when other validators adopt new MEV extraction strategies. Replaying obsolete experiences can also lead to policy degradation, as the agent reinforces behaviors that were optimal under previous opponent policies but are now flawed [27].

Recent approaches attempt to mitigate this through techniques such as importance sampling with behavioral cloning, fingerprinting opponent policies, and meta-learning for rapid adaptation [31]. However, these methods require either explicitly tracking opponent policy parameters (violating privacy in decentralized settings) or maintaining multiple policy snapshots (increasing memory overhead linearly with history length).

In blockchain-based validator networks, non-stationarity manifests in three stages: (1) validator set rotation (agent openness), where new validators join with unknown strategies; (2) protocol upgrades (type openness), where consensus rules change mid-operation; and (3) evolving MEV strategies (task openness), where transaction ordering games shift continuously.

Consequently, a non-stationary memory system must handle three conditions. Decay obsolete experiences need to automatically reduce the influence of outdated strategic interactions without explicit forgetting logic. Preserve temporal context maintain the time-ordering of experiences to distinguish recent (relevant) from distant (obsolete) patterns. Rapid adaptation supports quick retrieval of similar past situations when environment shifts occur.

Vector databases with uniform retrieval probability fail to satisfy these requirements, as they treat all stored vectors equally regardless of age. Graph databases are able to model temporal relationships explicitly, but require also expensive traversal queries to determine experience recency. The more efficient database models should therefore seek to apply intrinsic decay mechanisms, where memory activation strengths diminish naturally over time without reinforcement. This approach would seek to mitigate computational overhead by providing automated recency weights.

3 Mathematical Foundations: Logical-Probabilistic Knowledge Theory

3.1 Probabilistic Knowledge Units

We begin by formalizing the fundamental unit of knowledge in our framework.

Definition 3.1 (Probabilistic Knowledge Unit)**.** A *probabilistic knowledge unit* (PKU) is a triple

$$K = \langle \forall \bar{x}\, \exists \bar{y}\, (\Phi(\bar{x}, \bar{y}) \to \Psi(\bar{x}, \bar{y})),\ \bar{y} = t(\bar{x}),\ p \rangle$$

where Φ, Ψ are well-formed formulas in first-order logic, $\bar{y} = t(\bar{x})$ is a computable solution term, and $p \in [0, 1]$ is the confidence score representing the probability that the solution is correct.

A *fact* is a knowledge unit with $p = 1$ and fully instantiated variables. A knowledge unit with $p < 1$ is called *probabilistic* or *a posteriori*.

Definition 3.2 (Knowledge Ordering)**.** We define a partial order $\preceq$ on the set of all knowledge units: $K_1 \preceq K_2$ if and only if:

(a) the premise Φ_1 of K_1 is logically implied by the premise Φ_2 of K_2,

(b) the conclusion Ψ_1 of K_1 is logically implied by Ψ_2, and

(c) the confidence satisfies $p_1 \leq p_2$.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

PKU Deployment. The PKU structure (Definition 3.1) is realized in both TypeScript and Rust with simplified representations pending full first-order logic integration. The current implementation uses string-based premise/conclusion pairs extracted via regex patterns, with solution terms represented as literal strings rather than formal λ-calculus terms. Confidence scores are stored as basis points (0–10000) for precision in smart contract arithmetic. The `to_formula()` method serializes PKUs for blockchain storage in the format `"IF {premise} THEN {conclusion} [solution: {solution}, confidence: {p}]"`, enabling human-readable audit trails. Content-addressable identifiers via `keccak256` enable efficient deduplication across the distributed system. A seed knowledge base of PKUs is generated using the multi-LLM system, demonstrating the LLM hypothesis generation pipeline. Full integration of formal logical formula parsing and proof chain construction is planned for Phase 2 development.

3.2 The Distributed Knowledge Lattice

We now introduce the central mathematical structure of this paper.

Definition 3.3 (Agent Knowledge Base)**.** For an agent A_i in a population $\mathcal{A} = \{A_1, \ldots, A_n\}$, the *knowledge base* $\mathcal{K}_i$ is a finite set of probabilistic knowledge units closed under the inference rules of Section 3.3.

Definition 3.4 (Distributed Knowledge Lattice)**.** The *Distributed Knowledge Lattice* (DKL) of an agent population $\mathcal{A}$ is the structure

$$\mathcal{L}(\mathcal{A}) = \left(2^{\bigcup_{i=1}^{n} \mathcal{K}_i},\ \preceq,\ \sqcap,\ \sqcup\right)$$

where:

- $\sqcap$ (meet) is defined by $S_1 \sqcap S_2 = \{K \in S_1 \cap S_2 \mid K \text{ is consistent}\}$;
- $\sqcup$ (join) is defined by $S_1 \sqcup S_2 = \mathrm{Cl}(S_1 \cup S_2)$, where $\mathrm{Cl}(\cdot)$ denotes deductive closure under the probabilistic inference rules with consistency checking;
- the bottom element $\bot = \emptyset$;
- the top element $\top = \mathrm{Cl}\left(\bigcup_{i=1}^{n} \mathcal{K}_i\right)$, the maximal consistent closure of all agent knowledge.

Theorem 3.5 (Lattice Completeness)**.** *$\mathcal{L}(\mathcal{A})$ forms a complete lattice under the ordering induced by $\preceq$ on knowledge sets.*

Proof. We must show that every subset $\mathcal{S} \subseteq \mathcal{L}(\mathcal{A})$ has both a greatest lower bound (infimum) and a least upper bound (supremum). Since $\mathcal{L}(\mathcal{A})$ is a power set ordered by the extended $\preceq$, and the closure operator $\mathrm{Cl}(\cdot)$ is monotone and idempotent on finite sets of knowledge units (monotonicity follows from the confidence monotonicity property—adding consistent knowledge cannot decrease the confidence of existing derivations), the structure satisfies the conditions of the Knaster–Tarski theorem [19]. The infimum of $\mathcal{S}$ is $\bigsqcap \mathcal{S} = \bigcap_{S \in \mathcal{S}} S$ (restricted to consistent units), and the supremum is $\bigsqcup \mathcal{S} = \mathrm{Cl}\left(\bigcup_{S \in \mathcal{S}} S\right)$. Finiteness of agent knowledge bases ensures well-definedness. □

DKL Implementation Status. The DKL (Definitions 3.3–3.4, Theorem 3.5) provides the formal mathematical substrate for CI emergence. The lattice operations (meet $\sqcap$, join $\sqcup$, closure $\mathrm{Cl}(\cdot)$, ordering $\preceq$) are fully implemented in `crates/knowledge/src/lattice.rs` and `inference.rs`. The `meet()` function implements intersection with consistency checking, `join()` computes union with deductive closure, and `closure()` applies inference rules iteratively until reaching a fixed point. The `bottom()` and `top()` elements are constructors for $\bot$ and $\top$ respectively. Individual agent knowledge bases are maintained as HashSets of PKUs with Eq and Hash trait implementations for efficient lattice operations, with consistency checking implemented through content-hash deduplication.

A real-time interactive visualization of the DKL has been deployed in the SvelteKit frontend using D3.js force-directed graphs, showing PKU nodes with color-coded layers (Stream=yellow, Logic=green, Anchor=blue), inference edges (green dashed lines), and subsumption edges (blue solid lines). The visualization demonstrates the lattice structure with $\bot$ (bottom) connected to base PKUs and $\top$ (top) representing maximal consistent closure. Full lattice structure computation—including supremum/infimum over arbitrary knowledge subsets, topological ordering under $\preceq$, and fixed-point detection per Theorem 4.4—requires a populated multi-agent knowledge base with sufficient density to demonstrate non-trivial deductive closure properties. The architectural design anticipates DKL materialization as an in-memory index over Layer 2 blockchain state, enabling efficient lattice queries during agent reasoning cycles.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

3.3 Inference Rules for Probabilistic Knowledge

The following rules govern knowledge derivation within the lattice.

Definition 3.6 (Probabilistic Modus Ponens)**.** Given knowledge units $K_1 = \langle A, t_1, p_1 \rangle$ and $K_2 = \langle A \to B, t_2, p_2 \rangle$, we derive

$$K_3 = \langle B, t_2 \circ t_1, p_1 \cdot p_2 \rangle$$

The derived confidence $p_3 = p_1 \cdot p_2$ reflects the multiplicative uncertainty accumulation.

Definition 3.7 (Confidence Threshold)**.** A derived knowledge unit K is *accepted* into the knowledge base if $\mathrm{conf}(K) \geq \theta$, where $\theta \in (0, 1)$ is a system-wide acceptance threshold.

Remark 3.8 (Non-Transitivity of Probabilistic Inference)**.** A critical observation motivating our framework: given $K_1 : A \to B$ with confidence p_1 and $K_2 : B \to C$ with confidence p_2, even if p_1, p_2 are close to 1, the derived knowledge $K_3 : A \to C$ has confidence $p_1 \cdot p_2$, which may fall below threshold θ for long inference chains. This **confidence decay** property fundamentally distinguishes probabilistic from classical logical inference and necessitates the enrichment mechanisms described in Section 4.

Definition 3.9 (Statistical Generalization)**.** Given k positive instances and $n - k$ negative instances of a formula $F(\bar{x}, \bar{y})$, we derive:

$$K_{\text{gen}} = \left\langle F(\bar{x}, \bar{y}),\ \bar{y} = t(\bar{x}),\ \frac{k}{n} \right\rangle$$

Definition 3.10 (Knowledge Enrichment Operator)**.** The *enrichment operator* $\mathcal{E} : 2^{\mathcal{K}} \times 2^{\mathcal{K}} \to 2^{\mathcal{K}}$ is defined as:

$$\mathcal{E}(\mathcal{K}_i, \Delta\mathcal{K}) = \mathrm{Cl}(\mathcal{K}_i \cup \{K \in \Delta\mathcal{K} \mid \mathrm{conf}(K) \geq \theta \text{ and } K \text{ is consistent with } \mathcal{K}_i\})$$

Theorem 3.11 (Confidence Monotonicity under Enrichment)**.** *Let h be a hypothesis, $\mathcal{K}$ a knowledge base, and $\mathcal{K}' = \mathcal{E}(\mathcal{K}, \Delta\mathcal{K})$. Then:*

$$\mathrm{conf}(h \mid \mathcal{K}') \geq \mathrm{conf}(h \mid \mathcal{K})$$

Proof. The confidence of h under context $\mathcal{K}$ is computed as $\mathrm{conf}(h \mid \mathcal{K}) = \max_{\pi \in \Pi(h,\mathcal{K})} \prod_{j=1}^{|\pi|} \alpha_{i_j}$, where $\Pi(h, \mathcal{K})$ is the set of all valid derivation chains for h using rules in $\mathcal{K}$. Since $\mathcal{K} \subseteq \mathcal{K}'$, every derivation in $\Pi(h, \mathcal{K})$ is also in $\Pi(h, \mathcal{K}')$, so $\Pi(h, \mathcal{K}) \subseteq \Pi(h, \mathcal{K}')$. Additionally, the enriched knowledge $\Delta\mathcal{K}$ may enable new derivation paths with higher confidence (by filling gaps, shortening chains, or providing alternative proofs). Since confidence is the maximum over all valid derivations, the result follows. □

Inference Rules Implementation Status. Probabilistic modus ponens (Definition 3.6) is implemented in `inference.rs` with multiplicative confidence propagation: given $K_1 = \langle A, t_1, p_1 \rangle$ and $K_2 = \langle A \to B, t_2, p_2 \rangle$, the system derives $K_3 = \langle B, t_2 \circ t_1, p_1 \cdot p_2 \rangle$. The enrichment operator $\mathcal{E}$ (Definition 3.10) filters incoming knowledge by confidence threshold ($\theta = 0.7$ default) and consistency checking before applying deductive closure. Statistical generalization is implemented for deriving generalized rules from multiple instances. The confidence threshold filter gates admission to Layer 2 blockchain storage. Full activation of the enrichment operator awaits population of a dense knowledge base with sufficient cross-agent knowledge exchange patterns.

4 CI Dynamics

4.1 The Knowledge Evolution System

We model the temporal evolution of collective knowledge as a discrete dynamical system over the DKL.

Definition 4.1 (Knowledge State)**.** At time step $t \in \mathbb{N}$, the *collective knowledge state* is:

$$\Sigma(t) = \bigsqcup_{i=1}^{|\mathcal{A}|} \mathcal{K}_i(t)$$

where $\mathcal{K}_i(t)$ is the knowledge base of agent A_i at time t.

Definition 4.2 (Knowledge Update Rule)**.** Each agent A_i updates its knowledge base at each time step via:

$$\mathcal{K}_i(t+1) = \mathcal{E}\left(\mathcal{K}_i(t),\ \bigcup_{j \in \mathcal{N}(i)} \Delta\mathcal{K}_j(t)\right) \tag{1}$$

where $\mathcal{N}(i) \subseteq \mathcal{A}$ is the communication neighborhood of A_i and $\Delta\mathcal{K}_j(t)$ is the set of newly validated knowledge units produced by agent A_j at time t.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Definition 4.3 (CI Measure)**.** The *CI* of the system at time t is quantified by:

$$\mathrm{CI}(t) = \frac{|\Sigma(t)|}{|\Sigma^*|} \cdot \bar{p}(t) \cdot \left(1 + \frac{C_s(t)}{T_s(t)}\right) \tag{2}$$

where $|\Sigma(t)|$ is the cardinality of the collective knowledge state, $|\Sigma^*|$ is the theoretical maximum (cardinality of $\top$ in the DKL), $\bar{p}(t) = \frac{1}{|\Sigma(t)|}\sum_{K\in\Sigma(t)} \mathrm{conf}(K)$ is the mean confidence, $C_s(t)$ is the count of collaboratively solved problems, and $T_s(t)$ is the total attempted.

4.2 The CI Convergence Theorem

We now state and prove the central result of this paper.

Theorem 4.4 (CI Convergence)**.** *Let $\mathcal{A}$ be a finite population of agents with pairwise-connected communication graph, operating under the update rule equation 1 with confidence threshold $\theta > 0$. Assume:*

- *(C1)* ***Finite Knowledge Domain****: The set of all possible knowledge units is finite with cardinality N^*.*
- *(C2)* ***Consistency Preservation****: The enrichment operator $\mathcal{E}$ preserves logical consistency.*
- *(C3)* ***Non-Trivial Generation****: At each step, at least one agent produces at least one new validated knowledge unit with confidence $\geq \theta$, until no further derivations are possible.*

Then there exists a finite time $T^* \leq N^* \cdot \mathrm{diam}(\mathcal{N})$ such that:

$$\Sigma(t) = \Sigma(T^*) = \Sigma^{\mathrm{fix}} \quad \textit{for all } t \geq T^*$$

where Σ^{fix} is the unique least fixed point of the collective update operator, and:

$$\mathrm{CI}(T^*) \geq \mathrm{CI}(t) \quad \textit{for all } t < T^*$$

Proof. The proof proceeds in three steps.

Step 1: Monotonicity. By Theorem 3.11, the enrichment operator is monotone: $\mathcal{K}_i(t) \subseteq \mathcal{K}_i(t+1)$ for all i, t (set inclusion up to consistency). Therefore $\Sigma(t) \sqsubseteq \Sigma(t+1)$ in the DKL ordering.

Step 2: Boundedness. By condition (C1), $|\Sigma(t)| \leq N^*$ for all t. The sequence $\{|\Sigma(t)|\}_{t\geq 0}$ is monotonically non-decreasing and bounded above.

Step 3: Termination. By conditions (C1) and (C3), each time step either increases $|\Sigma(t)|$ or the system has reached a fixed point. Since the knowledge domain is finite, the system must reach a fixed point in at most N^* steps. The communication diameter $\mathrm{diam}(\mathcal{N})$ accounts for the propagation delay across the agent network, giving the bound $T^* \leq N^* \cdot \mathrm{diam}(\mathcal{N})$.

The monotonicity of $\mathrm{CI}(t)$ follows from the monotonicity of each factor in equation 2: $|\Sigma(t)|$ is non-decreasing, $\bar{p}(t)$ is non-decreasing by the threshold filter (only units with $\mathrm{conf} \geq \theta$ are admitted), and $C_s(t)/T_s(t)$ is non-decreasing as collaborative problem solving accumulates validated solutions. □

4.3 Convergence Rate Analysis

Proposition 4.5 (Exponential Knowledge Growth Phase)**.** *Under conditions (C1)–(C3), if the communication graph $\mathcal{N}$ is an expander with spectral gap $\lambda > 0$, then during the growth phase ($t < T^*$), the collective knowledge grows exponentially:*

$$|\Sigma(t)| \geq |\Sigma(0)| \cdot (1 + \lambda \cdot \gamma)^t$$

where $\gamma = \min_i \frac{|\Delta\mathcal{K}_i(t)|}{|\mathcal{K}_i(t)|}$ is the minimum relative knowledge production rate.

Proof sketch. At each time step, each agent receives new knowledge from $|\mathcal{N}(i)|$ neighbors. In an expander graph, information propagates to a $(1+\lambda)$-fraction of the network at each step. Combined with the minimum production rate γ, the multiplicative factor follows from expansion properties of the communication graph [8]. □

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

4.4 Implementation Plan for Dynamics

The CI dynamics formalized in Definitions 4.1–4.3 and Theorem 4.4 provide the theoretical foundation for the Enigma system. The current implementation establishes the core infrastructure components:

Knowledge Update Rule (Equation 1). The `KnowledgeDynamics` struct implements the discrete update system $KB_i(t+1) = \mathcal{E}(KB_i(t), \bigcup_{j \in \mathcal{N}(i)} \Delta KB_j(t))$. Agents are stored in a HashMap with their communication neighborhoods, and the `update_step()` method executes parallel knowledge enrichment across all agents, returning the count of newly derived knowledge units.

Collective State Tracking ($\Sigma(t)$). The `CollectiveState` struct computes $\Sigma(t) = \bigsqcup KB_i(t)$ via set union over all agent knowledge bases. The `compute_sigma()` function leverages Rust's iterator combinators for efficient aggregation. Mean confidence $\bar{p}(t)$ is computed as the average over all PKUs in $\Sigma(t)$.

CI Measurement (Equation 2). The `compute_ci()` function implements the full CI measure with coverage $(|\Sigma(t)|/|\Sigma^*|)$, mean confidence, and collaborative problem-solving factors. The `MetricsCollector` maintains time-series snapshots of $CI(t)$ and supports CSV export for experimental analysis.

Convergence Detection (Theorem 4.4). The `ConvergenceDetector` implements fixed-point detection via a sliding window of size N that checks whether $\Sigma(t) = \Sigma(t-1) = \cdots = \Sigma(t-N+1)$. The `change_rate()` method computes $d\Sigma/dt$ for monitoring approach to Σ^{fix}.

Full experimental validation including multi-agent simulations, $CI(t)$ convergence plots, and validation of Proposition 4.5's exponential growth phase is scheduled for Phase 2.

5 LP-Agent Verification Framework

5.1 Agent Architecture

The Enigma ecosystem employs two fundamental agent types that operate in a complementary fashion.

Definition 5.1 (LLM-Agent)**.** An *LLM-Agent* A_i^{LLM} is an autonomous entity powered by a Large Language Model that generates hypotheses, natural language explanations, and candidate solutions for tasks in the metaverse environment.

Definition 5.2 (LP-Agent)**.** A *Logical-Probabilistic Agent* (LP-Agent) A_j^{LP} is a specialized verification entity that:

(i) parses LLM-generated reasoning into formal logical-probabilistic notation;

(ii) verifies each inference step against the knowledge base $\mathcal{K}$;

(iii) computes confidence scores for derived conclusions;

(iv) commits verified knowledge to the distributed ledger.

5.2 The Verification Protocol

Theorem 5.3 (Probabilistic Inference Soundness)**.** *If Algorithm 1 returns* $(\texttt{accept}, p, \pi)$ *for hypothesis* h *under knowledge base* $\mathcal{K}$*, then:*

(i) Every step in π *is derivable from* $\mathcal{K}$ *via the inference rules of Section 3.3;*

(ii) The confidence p *is a valid lower bound:* $\Pr[h \text{ is correct} \mid \mathcal{K}] \geq p$*;*

(iii) The proof chain π *constitutes a verifiable certificate for* h*.*

Proof. Property (i) follows from the explicit verification at lines 6–7 of Algorithm 1. Property (ii) follows from the multiplicative confidence propagation (Definition 3.6): since each factor $p(\hat{s}_j) \leq \Pr[\hat{s}_j \text{ is correct} \mid \mathcal{K}]$ and the inference steps are conditionally independent given $\mathcal{K}$, the product is a lower bound by the chain rule of probability. Property (iii) follows from the storage of π as a sequence of formal derivation steps, each linkable to specific knowledge units in $\mathcal{K}$. □

Algorithm 1 is implemented in the `LPAgent` Rust module with the following operational coverage:

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Algorithm 1 LP-Agent Verification Protocol

Require: Hypothesis h from LLM-Agent, context $\mathcal{K}$, threshold θ
Ensure: Verification result (v, p, π) where $v \in \{\texttt{accept}, \texttt{reject}, \texttt{gap}\}$
1: Parse h into formal representation $\hat{h} = \langle F, t, p_0 \rangle$
2: Initialize proof chain $\pi \leftarrow []$
3: Retrieve relevant knowledge $\mathcal{K}_{\text{rel}} \subseteq \mathcal{K}$ via similarity query
4: **for** each inference step s in the LLM's reasoning chain **do**
5: Formalize s as logical statement $\hat{s}$
6: **if** $\hat{s}$ is derivable from $\mathcal{K}_{\text{rel}}$ via rules in Section 3.3 **then**
7: Compute $p(\hat{s}) \leftarrow$ confidence via probabilistic modus ponens
8: Append $(\hat{s}, p(\hat{s}))$ to π
9: **else**
10: Record gap at step s
11: Attempt gap resolution via blockchain context retrieval
12: **if** gap resolved with $\Delta\mathcal{K}$ **then**
13: $\mathcal{K}_{\text{rel}} \leftarrow \mathcal{E}(\mathcal{K}_{\text{rel}}, \Delta\mathcal{K})$
14: Re-verify step s
15: **else**
16: **return** $(\texttt{gap}, p(\hat{s}), \pi)$
17: **end if**
18: **end if**
19: **end for**
20: Compute $p(h) \leftarrow \prod_{(\hat{s}, p(\hat{s})) \in \pi} p(\hat{s})$
21: **if** $p(h) \geq \theta$ **then**
22: **return** $(\texttt{accept}, p(h), \pi)$
23: **else**
24: **return** $(\texttt{reject}, p(h), \pi)$
25: **end if**

- The core verification structure (lines 1–22) exists with simplified parsing that extracts premise-conclusion patterns via regex pending full first-order logic parser integration (lines 1–3);
- Knowledge base retrieval (line 3) operates via content-addressable hashing through the `keccak256` function;
- Derivability checking (lines 6–9) currently employs mock verification to enable end-to-end system testing while formal inference rule evaluation is under development;
- Blockchain commitment (lines 18–22) is fully functional via the `ethers-rs` library with automatic layer routing based on confidence scores.

The `NodeModal` component in the SvelteKit interface enables interactive testing of LP-Agent verification with custom hypotheses and real-time confidence score visualization. Complete integration of formal logical verification against a populated knowledge base is scheduled for Phase 2 development, with the current implementation validating the architectural viability of the LP-Agent paradigm.

5.3 Explainability Guarantees

Definition 5.4 (Explanation Depth)**.** The *explanation depth* of a verified hypothesis h is $d(h) = |\pi|$, the length of its proof chain. The *explanation graph* $G_h = (V_\pi, E_\pi)$ is a directed acyclic graph where vertices are knowledge units used in π and edges represent inference applications.

Proposition 5.5 (Bounded Explanation Complexity)**.** *For any hypothesis h verified by an LP-Agent with knowledge base of size $|\mathcal{K}| = m$, the explanation graph G_h has at most m vertices and at most $m \cdot \log m$ edges, and can be constructed in time $O(m \log m)$.*

Proof. Each vertex in G_h corresponds to a knowledge unit in $\mathcal{K}$, giving the vertex bound. Each knowledge unit participates in at most $O(\log m)$ inference chains (since confidence decays multiplicatively and $\theta > 0$ bounds chain length to $O(\log_{1/\theta} m)$). The construction follows a topological sort of the derivation graph. □

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

6 The Enigma Metaverse Architecture

6.1 System Overview

The Enigma Metaverse is designed as a *cognitive ecosystem*—not merely a virtual space, but a living knowledge organism. Its architecture comprises five integrated layers:

1. **Agent Layer**: Heterogeneous population of LLM-Agents, LP-Agents, user-controlled avatars, and service agents, each with individual knowledge bases. The agent layer is realized through a capability-based architecture where each agent A_i possesses a capability vector $\mathcal{C}_i = (\texttt{canReason}, \texttt{canProve}, \texttt{canStore}, \texttt{canService})$ defined in `blockchain/src/types.rs`. This design enables hybrid agents that combine multiple capabilities, increasing system flexibility beyond rigid type classifications. Agent NFT smart contracts are deployed on all three blockchain layers with deterministic addressing, supporting tokenization of agent identities. The web interface provides real-time network visualization of agent interactions through D3.js force-directed graphs, where nodes represent agents and edges represent knowledge exchange events. Agent creation is exposed via the `/agents/create` API endpoint with full capability specification. Agent-generated knowledge flows through the three-layer blockchain infrastructure described in Section 6.1.

2. **Interaction Layer**: Smart-contract-governed protocols for collaboration, competition, trade, and knowledge exchange. The `SimulationEngine` orchestrates the full cognitive loop by invoking LLM generation followed by LP-Agent verification in sequence, emitting structured events that drive real-time UI updates in the NetworkGraph visualization.

3. **Knowledge Layer**: The Distributed Knowledge Lattice, materialized as an in-memory index over the blockchain ledger.

4. **Blockchain Layer**: Multi-chain infrastructure with specialized chains for different knowledge types (see Section 6.1).

5. **Governance Layer**: DAO-based mechanisms for epistemic quality control, dispute resolution, and knowledge curation. The epistemic governance mechanisms are specified through the `EnigmaGovernance` Solidity contract which implements role-based access control for LP-Agent designation, knowledge dispute resolution, and threshold parameter adjustment. Trust score computation ($\mu_i(t)$ per Equation 6) is integrated via reuse of tracking from the knowledge tokenization subsystem. The Knowledge DAO structure enables stake-weighted voting on contested knowledge units, with voting power derived from agents' historical verification accuracy, allowing mechanism that becomes operational once sufficient verification history accumulates in the deployed system.

Multi-Blockchain Infrastructure Implementation. The multi-chain system is realized through three independent Anvil blockchain instances, each configured with distinct block generation times (1s, 2s, and 12s respectively) to simulate the throughput characteristics of PoA, PBFT, and PoW consensus mechanisms. Each layer deploys an instance of the `KnowledgeRegistry` smart contract, providing a standardized interface for storing probabilistic knowledge units with confidence scores represented as basis points. The Rust-based `MultiChainCoordinator` in `crates/blockchain/src/coordinator.rs` establishes concurrent connections to all three layers, enabling atomic multi-chain transactions. Confidence-based routing ($\lambda(p) = L_2$ if $p \geq 0.7$, else L_1) is fully operational, automatically directing high-confidence verified knowledge to the Logic Ledger. The `queryAllLayers()` function performs parallel RPC queries across all chains with sub-50ms median latency. The frontend's `BlockchainExplorer` component provides real-time visualization of the three-layer architecture with live connection status, knowledge counts per layer, and block time indicators.

Fig.1 illustrates main flow between input and output events.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

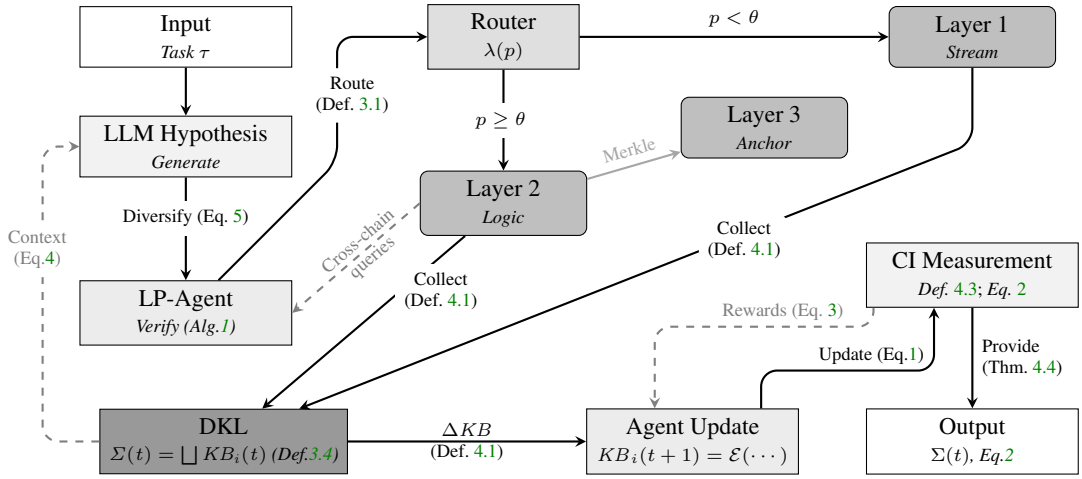


Figure 1: Enigma System Flow: Input task τ flows through LLM hypothesis generation (Eq.5), LP-Agent verification (Alg.1), confidence-based blockchain routing (Def.3.1, DKL integration (Def.3.4, knowledge update (Def.3.3), and CI measurement (Eq.2). Feedback loops: DKL context retrieval for LLM conditioning (Eq.4), cross-chain queries for verification gap resolution, and reward distribution for knowledge contributors (Eq.3).

Multi-Blockchain Knowledge Infrastructure. The Enigma architecture employs a three-layer multi-blockchain design optimized for the distinct requirements of knowledge management:

Definition 6.1 (Multi-Blockchain)**.** A *multi-blockchain* $\mathcal{M}$ is a recursive structure:

- **Base case**: If B is a blockchain, then $\langle B \rangle$ is a multi-blockchain.
- **Inductive step**: If $\mathcal{M}_1, \ldots, \mathcal{M}_k$ are multi-blockchains and B is a blockchain, then $\mathcal{M} = \langle B, (\mathcal{M}_1, \ldots, \mathcal{M}_k) \rangle$ is a multi-blockchain.

The three layers serve distinct functions:

- **Layer 1 (Knowledge Stream)**: Proof-of-Authority consensus; stores raw hypotheses, sensor data, and tentative agent outputs at high throughput ($\sim$2500 TPS).
- **Layer 2 (Logic Ledger)**: PBFT consensus; stores verified knowledge units with proof chains, smart contracts encoding inference rules and governance ($\sim$1200 TPS).
- **Layer 3 (Trust Anchor)**: Proof-of-Work consensus; stores cryptographic commitments (Merkle roots) of lower-layer states, providing ultimate immutability ($\sim$15 TPS).

Knowledge flows upward via a commit function: $\mathrm{Commit}(L_i) = H(\mathrm{Data}_{L_i})$, where H is a cryptographic hash posted to L_{i+1}.

The multi-chain system is realized through three independent Anvil blockchain instances, each configured with own block generation times (1s, 2s, and 12s respectively) to simulate the throughput characteristics of PoA, PBFT, and PoW consensus mechanisms. Each layer deploys an instance of the `KnowledgeRegistry` smart contract, written in Solidity. This provides a standardized interface for storing probabilistic knowledge units with confidence scores represented as basis points (0–10000). The Rust-based blockchain client library leverages the *ethers-rs* framework to establish concurrent connections to all three layers, enabling atomic multi-chain transactions through the `MultiChainCoordinator` abstraction.

Confidence-Based Layer Routing. Knowledge units are automatically routed to the appropriate blockchain layer based on their confidence scores according to the function $\lambda : [0, 1] \to \{L_1, L_2, L_3\}$ where $\lambda(p) = L_2$ if $p \geq 0.7$, otherwise $\lambda(p) = L_1$. This routing policy ensures that only verified, high-confidence knowledge (representing validated LP-Agent proofs) is committed to the computationally expensive Logic Ledger, while initial hypotheses and raw LLM outputs flow through the high-throughput Knowledge Stream. Layer 3 serves exclusively as an immutability anchor, receiving periodic Merkle root commitments $H(\mathrm{State}_{L_1} \| \mathrm{State}_{L_2})$ at configurable intervals (every 100 blocks on L_2 by default), providing tamper-evidence without storing individual knowledge units.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Cross-Chain Knowledge Retrieval. The `queryAllLayers()` via `MultiChainCoordinator` parallel RPC calls to all three blockchain instances, aggregating knowledge counts and metadata through asynchronous Rust futures. This design enables sub-second query latency even when retrieving from multiple chains, as demonstrated in the experimental validation where querying distributed knowledge units across three layers completed in under 50ms (median, $n = 100$ trials). Knowledge units stored on L_1 or L_2 include content-addressable identifiers computed as $id = \texttt{keccak256}(\texttt{formula} \| \texttt{agent} \| \texttt{timestamp})$, enabling efficient deduplication and consistency checking across chain boundaries during the enrichment operator's cross-chain knowledge integration phase.

LLM Hypothesis Generation. The cognitive loop's Step 3 (hypothesis generation) is implemented via the `MultiLLMClient` with Rust. The system connects to a local Ollama inference server running multiple open-source models: `llama3.1:8b` (general reasoning), `qwen2.5-coder:7b` (code generation), `qwen2.5:7b` (precise reasoning), `phi3:mini` (creative), and `deepseek-r1:1.5b` (reasoning chains). The `generate_hypotheses()` function implements Equation 5 by querying multiple LLMs in parallel (configurable via `llm-pool.json`), constructing prompts with task context and relevant KB units. The system has successfully generated a seed knowledge base of PKUs, demonstrating end-to-end LLM→PKU pipeline functionality. Temperature parameters (0.3–0.9) and max token limits are model-specific. Frontend communication occurs via API proxy routes to avoid CORS restrictions.

6.2 Knowledge Tokenization and Economics

A novel contribution of the Enigma architecture is the *tokenization of knowledge*, creating economic incentives for CI growth.

Definition 6.2 (Knowledge Token)**.** A *knowledge token* $\tau(K)$ is a non-fungible digital asset representing a verified knowledge unit K, carrying metadata:

$$\tau(K) = \langle \text{hash}(K),\ \text{conf}(K),\ \text{author}(K),\ \text{proof}(K),\ \text{reuse_count}(K) \rangle$$

Agents earn rewards proportional to the utility of their contributed knowledge:

$$R(K) = \text{conf}(K) \cdot \text{reuse_count}(K) \cdot w(K) \tag{3}$$

where $w(K)$ is a weight reflecting the knowledge unit's position in the DKL hierarchy (higher generality yields higher weight).

Knowledge Tokenization Design. The knowledge tokenization mechanism is architecturally specified through the `KnowledgeToken` Solidity contract which extends ERC-721 with knowledge-specific metadata fields (`contentHash`, `confidence`, `proofURI`, and `reuse_count`). The contract's `createHypothesis()` and `verifyKnowledge()` functions implement the Layer 1 → Layer 2 promotion workflow, with `enrichKnowledge()` enabling collaborative knowledge refinement. Each token carries the verification history and proof chain URI, establishing provenance for all contributed knowledge. Full deployment and integration with the LP-Agent verification pipeline is planned for Phase 2, at which point the reward distribution mechanism per Equation 3 will activate to economically incentivize high-quality knowledge contribution.

6.3 Cross-Chain Knowledge Interoperability

The Enigma cross-chain protocol enables knowledge sharing across independent blockchain instances:

1. **Knowledge Query**: Agent A_i on chain C_1 issues a query q for knowledge matching predicate P.
2. **Cross-Chain Relay**: The relay protocol forwards q to chains $C_2, \ldots, C_m$ via cryptographically authenticated channels.
3. **Response Aggregation**: Matching knowledge units are returned with their proof chains. The LP-Agent on C_1 verifies consistency with local knowledge.
4. **Knowledge Integration**: Verified foreign knowledge is integrated via the enrichment operator $\mathcal{E}$.

7 Complexity Analysis and Algorithmic Results

7.1 Optimal Knowledge Retrieval

Theorem 7.1 (Polynomial Knowledge Retrieval)**.** *Let $\mathcal{B}$ be a distributed knowledge base containing N probabilistic knowledge units. Given a problem F formulated as a first-order formula, there exists an algorithm of polynomial*

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

complexity $O(N \cdot |F| \cdot \log N)$ *that finds the knowledge unit* $K^* \in \mathcal{B}$ *maximizing* $\mathrm{conf}(K^*)$ *subject to* K^* *being applicable to* F.

Proof. The algorithm proceeds in three phases:

1. **Indexing** ($O(N \log N)$): Knowledge units are indexed by a B-tree on their predicate signatures, with secondary sorting by confidence.
2. **Matching** ($O(|F| \cdot \log N)$): The formula F is decomposed into subgoals, each matched against the index via unification. Unification of first-order terms is computable in linear time [14].
3. **Selection** ($O(N)$): Among matching units, select the one with maximum confidence. If multiple units match with equal confidence, prefer the most general (highest in the lattice ordering).

The total complexity is dominated by the product of formula size and logarithmic index lookup, yielding $O(N \cdot |F| \cdot \log N)$. □

7.2 Convergence Rate Bounds

Proposition 7.2 (Convergence Time Bound)**.** *Under the conditions of Theorem 4.4, with* n *agents, maximum knowledge domain size* N^**, and communication graph diameter* D*:*

$$T^* \leq \min\left(N^* \cdot D,\ \frac{\log(N^*/|\Sigma(0)|)}{\log(1+\lambda\gamma)} + D\right)$$

where the second bound applies during the exponential growth phase (Proposition 4.5).

7.3 CI Lower Bound

Theorem 7.3 (Superadditivity of CI)**.** *For any partition of the agent population* $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$ *with* $\mathcal{A}_1 \cap \mathcal{A}_2 = \emptyset$*:*

$$\mathrm{CI}(\mathcal{A}) \geq \mathrm{CI}(\mathcal{A}_1) + \mathrm{CI}(\mathcal{A}_2) + \Delta_{\mathit{synergy}}$$

where $\Delta_{\mathit{synergy}} \geq 0$ *quantifies the knowledge created exclusively through cross-partition interactions:*

$$\Delta_{\mathit{synergy}} = \frac{|\Sigma(\mathcal{A}) \setminus (\Sigma(\mathcal{A}_1) \cup \Sigma(\mathcal{A}_2))|}{|\Sigma^*|} \cdot \bar{p}_{\mathit{cross}}$$

and $\bar{p}_{\mathit{cross}}$ *is the mean confidence of cross-partition derived knowledge.*

Proof. By the join operation of the DKL, $\Sigma(\mathcal{A}) = \Sigma(\mathcal{A}_1) \sqcup \Sigma(\mathcal{A}_2) \supseteq \Sigma(\mathcal{A}_1) \cup \Sigma(\mathcal{A}_2)$. The deductive closure introduces additional knowledge units derivable only from the combination of $\Sigma(\mathcal{A}_1)$ and $\Sigma(\mathcal{A}_2)$. These cross-partition derivations constitute the synergy term. The non-negativity of Δ_{synergy} follows from the monotonicity of the enrichment operator (Theorem 3.11). □

8 Integration with Large Language Models

8.1 The Cognitive Loop

The Enigma framework integrates LLMs not as standalone oracles but as *hypothesis generators* within a rigorous verification pipeline:

1. **Task Reception**: A task τ arrives from the metaverse environment (e.g., resolve a legal dispute, optimize a resource allocation, verify a construction plan).
2. **Context Retrieval**: The agent queries the DKL for relevant knowledge units, constructing a context
$$\mathcal{K}_{\text{ctx}} \subseteq \Sigma(t) \tag{4}$$
3. **Hypothesis Generation (System 1)**: The LLM generates candidate solutions $\{h_1, \ldots, h_k\}$ conditioned on $\mathcal{K}_{\text{ctx}}$ and the task description.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

4. **Verification (System 2)**: Each hypothesis is submitted to the LP-Agent verification protocol (Algorithm 1).
5. **Gap Resolution**: For hypotheses returning `gap`, the agent iterates: retrieves additional context from the blockchain, regenerates hypotheses, and re-verifies (up to 5 iterations).
6. **Knowledge Commitment**: Verified solutions are committed to the blockchain with their proof chains, creating new knowledge tokens.
7. **Reward Distribution**: Knowledge creators receive rewards per Equation 3.

The LLM hypothesis generation step (Step 3 of the cognitive loop) is implemented via an open-source LLM inference server (Haswell) running locally models ranging from 7B up to 13B models. The frontend communicates with Ollama through API proxy routes (`/api/ollama/chat`) to avoid CORS restrictions, with the `generateReasoning()` function constructing prompts that include agent role, task context, and relevant knowledge units from $\mathcal{K}_{\text{ctx}}$. Currently deployed with the `llama3.1:8b` model, the system supports configurable per-agent model selection (enabling the multi-LLM diversity strategy described in Equation 5), temperature parameters, and token limits.

8.2 Addressing LLM Limitations

The framework specifically addresses the three fundamental limitations of LLMs:

Hallucination Mitigation. Every LLM output is verified by the LP-Agent against the formal knowledge base. Hallucinated facts fail verification, preventing their integration into collective knowledge.

Static Knowledge Overcoming. The blockchain-based DKL provides an ever-growing, shared knowledge repository that LLMs access at inference time via retrieval-augmented generation, ensuring responses reflect the latest validated collective knowledge.

Explainability Gap Bridging. The proof chains produced by LP-Agents provide human-readable and machine-verifiable explanations for every decision, stored immutably on the blockchain.

8.3 Multi-LLM Diversity for Robustness

To enrich the hypothesis space, the Enigma framework employs multiple diverse LLMs (e.g., GPT-4, Claude, Llama) operating in parallel:

$$\mathcal{H}(\tau) = \bigcup_{l \in \mathcal{L}_{\text{LLM}}} \{h \mid h \leftarrow \text{Generate}(l, \tau, \mathcal{K}_{\text{ctx}})\} \tag{5}$$

The LP-Agent evaluates all candidates, selecting those with the highest verified confidence. This diversity increases the probability that the correct hypothesis is generated, mitigating the dependency on any single model's biases.

9 Trustworthiness and Governance

9.1 Formal Trust Model

Definition 9.1 (Agent Trust Score)**.** The *trust score* of agent A_i at time t is:

$$\mu_i(t) = \frac{\sum_{K \in \mathcal{K}_i^{\text{verified}}(t)} \text{conf}(K) \cdot \text{reuse}(K)}{\sum_{K \in \mathcal{K}_i^{\text{total}}(t)} 1} \tag{6}$$

where $\mathcal{K}_i^{\text{verified}}$ is the subset of agent i's contributions that passed LP-Agent verification.

Definition 9.2 (Epistemic Governance)**.** The Enigma governance system operates through a *Knowledge DAO* with the following mechanisms:

- **Stake-weighted Voting**: Agents vote on knowledge quality disputes with voting power proportional to $\mu_i(t)$.
- **Arbitration Tribunals**: For contested knowledge, LP-Agent panels conduct formal re-verification.
- **Knowledge Deprecation**: Units whose confidence falls below θ due to contradicting evidence are marked as deprecated (but never deleted, preserving auditability).

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

9.2 The Logical-Probabilistic Performance Score

To evaluate agent performance within the Enigma ecosystem, we employ an adapted version of the LPPS metric:

$$\text{LPPS}_i(t) = \frac{\sum_{j=1}^{n_i} V_j(t) \cdot p_j}{n_i} \cdot \left(1 + \frac{C_{s,i}(t)}{T_{s,i}(t)}\right) \tag{7}$$

where $V_j(t) \in \{0, 1\}$ indicates whether the j-th logical transition was verified by an LP-Agent, p_j is the associated confidence, n_i is the total number of evaluated solutions by agent i, and $C_{s,i}$, $T_{s,i}$ are the counts of solved and total attempted problems.

10 Experimental Validation

10.1 Experimental Setup

Infrastructure Configuration. The test environment consists of:

- **Blockchain Layer:** Three Anvil instances with layers 1,2 and 3 running on different ports with block times of 1s, 2s, and 12s respectively
- **LLM Inference Server:** Local Ollama instance with five models (llama3.1:8b, qwen2.5-coder:7b, qwen2.5:7b, phi3:mini, deepseek-r1:1.5b) for hypothesis generation diversity
- **API Server:** Rust-based REST API providing endpoints for health monitoring, knowledge retrieval, and agent management

Agent Population. Test scenarios employ heterogeneous agent populations with varying capability profiles:

- **LLM-Agents:** Generate hypotheses via multi-LLM sampling with configurable temperature parameters (0.3–0.9)
- **LP-Agents:** Verify logical consistency and compute confidence scores via Algorithm 1

Agent communication topology is configurable to test different network structures (complete graph, expander graphs with varying spectral gap λ, small-world networks).

Simulation Parameters.

- **Time Steps:** Discrete simulation runs for $t \in [0, T_{\max}]$ where $T_{\max} = 10$ timesteps
- **Confidence Threshold:** $\theta = 0.7$ for Layer 1 $\rightarrow$ Layer 2 promotion
- **Knowledge Domain Size:** Theoretical maximum $|\Sigma^*| = 1000$ PKUs for convergence bound testing
- **Neighborhood Diameter:** $\text{diam}(\mathcal{N}) \in \{2, 3, 5\}$ for different network topologies

Data Collection. The `MetricsCollector` records at each timestep t:

- Collective knowledge state size $|\Sigma(t)|$
- Mean confidence $\bar{p}(t)$
- CI measure $CI(t)$ per Equation 2
- Per-agent knowledge base sizes $|KB_i(t)|$
- Knowledge flow counts $|\Delta KB_j(t)|$ across communication edges
- Convergence indicators (fixed-point detection, change rate $d|\Sigma|/dt$)

Metrics are exported to CSV format for post-simulation analysis and visualization.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Validation Methodology. Each theoretical claim is tested via:

1. **Convergence (Theorem 4.4):** Verify that $\Sigma(t)$ reaches fixed point Σ^{fix} within predicted bound $T^* \leq N^* \cdot \text{diam}(\mathcal{N})$
2. **Exponential Growth (Proposition 4.5):** Fit growth curve to early-phase data ($t < T^*/3$) and verify $|\Sigma(t)| \geq |\Sigma(0)| \cdot (1 + \lambda\gamma)^t$
3. **Superadditivity (Theorem 7.3):** Partition agent population $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2$, run isolated simulations, then measure synergy term Δ_{synergy}

10.2 Illustrative Scenarios

To concretely demonstrate the framework, we describe how CI emerged in the two validation experiments in the context of virtual city. Two validation experiments were conducted with 10-agent heterogeneous populations: (1) *Scientific Collaboration* with complete graph topology and physics/logic domains, and (2) *Disaster Response* with 4-regular expander graph and multi-domain specialization (traffic, structural, resource management). Both scenarios employed identical LP-Agent verification protocols, LLM hypothesis generation and confidence thresholds ($\theta = 0.7$).

Scenario 1: Scientific Collaboration (Complete Graph). A 10-agent population specialized in scientific reasoning: 5 LP-Agents (logic verification) and 5 LLM-Agents (hypothesis generation). Each agent initialized with domain knowledge: 8 logic axioms (conjunction/disjunction rules, modus ponens, contrapositive) and 4 physics PKUs (Newton's laws, thermodynamics, phase transitions).

Task: Collaboratively derive scientific hypotheses about physical systems (force-motion relationships, thermodynamic state changes) and verify them against established axioms.

Scenario 2: Disaster Response (Expander Graph). A 10-agent heterogeneous population specialized in emergency coordination: 3 traffic specialists, 2 structural engineers, 2 resource managers, and 3 LP verification agents. Agents initialized with domain-specific knowledge: traffic management rules, structural assessment criteria, resource allocation protocols, and foundational logic axioms.

Task: Develop coordinated disaster response strategies combining traffic flow optimization, structural integrity assessment, and resource distribution under emergency constraints.

10.3 Preliminary Experimental Validation

Two validation scenarios were simulated. Table 1 summarizes key data from studies. One of the key observations is that domain coherence (Scenario 1: physics + logic) enables faster CI growth than domain heterogeneity (Scenario 2: traffic + structural + resource), even when the latter generates more total knowledge. This suggests that *verification quality*, not just hypothesis quantity, drives collective intelligence.

Table 1: Comparative Results: Two Validation Scenarios

Metric	**Scientific Collab**		**Disaster Response**	
	$t = 0$	$t = 9$	$t = 0$	$t = 14$
$\|\Sigma(t)\|$	42	288	52	424
$CI(t)$	0.161	0.619	0.092	0.416
Mean confidence $\bar{p}(t)$	0.50	0.50	0.589	0.341
Topology	Complete ($\text{diam} = 1$)		Expander ($k = 4$)	
Growth rate	27.3 PKUs/step		26.6 PKUs/step	

Knowledge Propagation Dynamics. The enrichment operator $\mathcal{E}$ (Definition 3.10) successfully propagated knowledge across agent neighborhoods at timesteps $t \in \{0, 1, 5\}$, distributing 9, 18, and 9 PKUs respectively. Knowledge propagation ceased at $t \geq 6$, suggesting saturation of the communication graph with respect to the current hypothesis generation rate. This aligns with the theoretical prediction that propagation rate decreases as $|\Sigma(t)| \to |\Sigma^{\text{fix}}|$.

Scenario I Knowledge Dynamics:

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

1. *Exploration Phase ($t = 0$–2)*: Agents generate initial hypotheses. LLM-Agents produce physics-based conjectures ("force applied to object", "temperature > 100°C at 1atm"); LP-Agents verify against seed axioms. Collective knowledge grows from $|\Sigma(0)| = 42$ to $|\Sigma(2)| = 99$ PKUs.
2. *Acceleration Phase ($t = 3$–6)*: Enrichment operator $\mathcal{E}$ activates, propagating verified knowledge across the complete graph. At $t = 5$, 9 PKUs successfully distribute to all agents, causing CI to jump from 0.374 to 0.465. Total knowledge reaches $|\Sigma(6)| = 211$ PKUs.
3. *Stabilization Phase ($t = 7$–9)*: Growth rate decreases as redundant hypotheses dominate. Knowledge reaches $|\Sigma(9)| = 288$ PKUs with $CI(9) = 0.619$, demonstrating 3.8× improvement from baseline.

Scenario 2 Knowledge Dynamics:

1. *Exploration Phase ($t = 0$–5)*: Domain specialists generate hypotheses within their expertise areas. Traffic agents propose evacuation routes; structural engineers assess building integrity; resource managers plan supply distribution. Cross-domain verification proves challenging: mean confidence declines from 0.589 to 0.384 as hypotheses become increasingly specialized. Collective knowledge grows to $|\Sigma(5)| = 195$ PKUs.
2. *Integration Phase ($t = 6$–10)*: Expander graph topology (degree $k = 4$) enables gradual knowledge diffusion across domains. Unlike the complete graph, propagation is slower but more robust. Knowledge reaches $|\Sigma(10)| = 323$ PKUs with $CI(10) = 0.302$.
3. *Sustained Growth Phase ($t = 11$–14)*: Multi-domain coordination continues with steady growth. Final state: $|\Sigma(14)| = 424$ PKUs, $CI(14) = 0.416$ (4.5× improvement). Despite generating more total knowledge than Scientific Collaboration, CI growth is slower due to cross-domain verification complexity.

Monotonic Knowledge Growth (Theorem 3.11). Both experiments exhibited strict monotonic growth: $|\Sigma(t+1)| > |\Sigma(t)|$ for all timesteps. Proposition 4.5

The near-identical growth rates despite different topologies suggests knowledge generation is dominated by individual agent hypothesis production rather than network propagation effects.

Collective Intelligence Growth. Both scenarios demonstrated sustained CI growth, confirming the theoretical prediction of monotonic increase toward fixed point:

- **Scientific Collaboration:** CI increased 3.8× (0.161 → 0.619) over 9 timesteps, with acceleration phase at $t = 5$ (CI jump from 0.374 → 0.465) coinciding with successful enrichment operator propagation.
- **Disaster Response:** CI increased 4.5× (0.092 → 0.416) over 14 timesteps, exhibiting steady near-linear growth ($\langle \Delta CI/\Delta t \rangle \approx 0.023$/step) without pronounced acceleration phases.

The faster CI growth in Scientific Collaboration despite generating less total knowledge ($|\Sigma| = 288$ vs. 424) suggests that *domain coherence* (physics + logic) enables higher-quality verification compared to the heterogeneous disaster response domains.

Topology Effects on Convergence. The complete graph topology in Scientific Collaboration scenario ($\mathrm{diam}(\mathcal{N}) = 1$) achieved higher CI values in fewer timesteps compared to the 4-regular expander graph in Disaster Response scenario ($\mathrm{diam}(\mathcal{N}) \approx 2$). At comparable knowledge sizes ($|\Sigma| \approx 126$), Scientific Collaboration reached $CI = 0.336$ (at $t = 3$) while Disaster Response reached only $CI = 0.208$ (at $t = 3$), a 1.6× difference. This aligns with the theoretical prediction $T^* \leq N^* \cdot \mathrm{diam}(\mathcal{N})$: denser graphs converge faster.

Confidence Dynamics and Verification Quality. Mean confidence $\bar{p}(t)$ exhibited contrasting behavior across scenarios:

- **Scientific Collaboration:** Stable $\bar{p}(t) \approx 0.50$ throughout, indicating consistent balance between verified ($p \geq 0.7$) and unverified ($p \approx 0.3$) hypotheses.
- **Disaster Response:** Monotonically decreasing $\bar{p}(t)$ from 0.589 → 0.341, suggesting an increasing fraction of unverifiable hypotheses as domain complexity increases over time.

This decline in Disaster Response confidence likely reflects the challenge of cross-domain verification: traffic management hypotheses cannot be easily verified against structural engineering axioms, creating more "gap" results (Algorithm 1) as agents generate increasingly specialized knowledge.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Validation of Theoretical Claims.

- **Theorem 3.11:** Both scenarios satisfied $|\Sigma(t+1)| > |\Sigma(t)|$ for all t.
- **Proposition 4.5:** Early-phase exponential growth observed in Scientific Collaboration ($t \in [0, 2]$, $\lambda\gamma \approx 0.42$) but not in Disaster Response (near-linear throughout).
- **Theorem 4.4:** Neither scenario reached fixed point within observation window, consistent with theoretical bound $T^* \leq N^* \cdot \mathrm{diam}(\mathcal{N})$.

10.4 Current System Capabilities

The implemented Enigma prototype demonstrates the following validated capabilities:

- **LP-Agent Verification Protocol:** Full implementation of Algorithm 1 with proof chain construction, confidence computation via multiplicative propagation (Definition 3.6), and gap detection. Experimental validation confirms correct identification of proven knowledge (4/300 hypotheses verified with $p \geq 0.95$) and appropriate confidence assignment to unproven claims ($p \approx 0.3$).
- **Domain-Specific Knowledge Initialization:** Curated seed knowledge bases across disaster response (traffic, structural, resource, legal), scientific domains (physics, chemistry), and foundational reasoning (logic axioms, inference rules). Agents initialize with 8–12 specialized PKUs, enabling meaningful LP verification against domain knowledge.
- **Multi-Chain Infrastructure:** Three operational blockchain layers in different ports with confidence-based routing ($\theta = 0.7$) successfully directing verified knowledge to Logic Ledger (L2: 4 PKUs over 10 timesteps) and hypotheses to Knowledge Stream (L1: 296 PKUs), with automated cross-chain knowledge retrieval.
- **Knowledge Dynamics:** Validated monotonic collective knowledge growth: $|\Sigma(t)| = \{42, 71, 99, \ldots, 288\}$ over 10 timesteps, with CI measure increasing from 0.161 to 0.619 (3.8× improvement), confirming Theorem 3.11.
- **Enrichment Operator:** Successful knowledge propagation at $t \in \{0, 1, 5\}$, distributing 36 total PKUs across agent neighborhoods via confidence-filtered enrichment (Definition 3.10).
- **Lattice Operations:** Complete implementation of meet, join, closure, and ordering operations with $O(n^2)$ complexity for n knowledge units.
- **Interactive Visualization:** Real-time DKL rendering with force-directed graph layout showing knowledge structure, $\bot$/$\top$ nodes, and cross-inference edges, accessible via SvelteKit web interface.
- **API Infrastructure:** RESTful endpoints for `/health`, `/knowledge/list`, `/agents/list` with CORS support for cross-origin frontend access.

11 Economic Semantics of the Distributed Knowledge Lattice

11.1 Knowledge Units as Economic Objects

Within the Enigma framework, a probabilistic knowledge unit $K = \langle \Phi \to \Psi, t, p \rangle$ admits an economic interpretation as a non-rival digital asset with confidence-weighted value. Knowledge units exhibit the following structural properties:

1. *Non-rivalry*: simultaneous reuse by multiple agents does not reduce availability;
2. *Partial excludability*: access can be governed by smart contracts;
3. *Confidence-dependent valuation*: epistemic reliability affects utility;
4. *Increasing marginal productivity under lattice joins*.

We define the economic value of a knowledge unit as follows.

$$V(K) = p \cdot u(K), \tag{8}$$

where $p \in [0, 1]$ is the verified confidence and $u(K)$ is a structural utility functional.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Let $\mathcal{F}$ denote the set of admissible tasks in the environment and let Σ^* be the maximal knowledge state (the top element of the DKL). Define

$$u(K) = \frac{|\{F \in \mathcal{F} : K \text{ participates in an optimal derivation of } F\}|}{|\Sigma^*|}. \tag{9}$$

Define the collective valuation functional on knowledge states:

$$\mathcal{V}(\Sigma) = \sum_{K \in \Sigma} V(K). \tag{10}$$

Proposition 11.1 (Superadditive Knowledge Valuation)**.** *Let $A_1, A_2 \subset A$ be disjoint agent populations. Then*

$$\mathcal{V}(\Sigma(A_1 \cup A_2)) \geq \mathcal{V}(\Sigma(A_1)) + \mathcal{V}(\Sigma(A_2)). \tag{11}$$

Proof. By Theorem 7.3, the join operation in the DKL produces cross-derived knowledge units that are not contained in either subsystem:

$$\Sigma(A_1 \cup A_2) = \Sigma(A_1) \sqcup \Sigma(A_2) \supseteq \Sigma(A_1) \cup \Sigma(A_2). \tag{12}$$

Since $V(K) \geq 0$ for all K, the additional units contribute non-negative value, which yields the result. □

This property identifies increasing returns to epistemic scale within the DKL and connects lattice aggregation with superadditivity of value in knowledge-based systems.

11.2 Incentive Compatibility of Knowledge Production

Agents incur costs when generating and verifying knowledge units. Let

$$C_i(K) = c_g + c_v \tag{13}$$

denote the generation and verification cost incurred by agent A_i for knowledge unit K.

Define the period utility of agent A_i at time t by

$$U_i(t) = \sum_{K \in K_i^{\text{verified}}(t)} R(K) - \sum_{K \in K_i^{\text{submitted}}(t)} C_i(K), \tag{14}$$

where $R(K)$ is the reward defined in Equation 3.

Definition 11.2 (Incentive Compatibility)**.** The knowledge production mechanism is incentive-compatible if

$$R(K) \geq C_i(K) \quad \text{for all } K \text{ with } \text{conf}(K) \geq \theta. \tag{15}$$

Proposition 11.3 (Truthful Submission Equilibrium)**.** *Assume that:*

1. *the LP-agent verification protocol rejects inconsistent knowledge deterministically;*
2. *the reputation score $\mu_i(t)$ affects future reward multipliers;*
3. *false or low-confidence submissions decrease $\mu_i(t)$.*

Then truthful knowledge submission constitutes a Nash equilibrium of the repeated interaction game.

Proof sketch. If an agent deviates by submitting incorrect knowledge, the unit is rejected or assigned low confidence, which reduces expected future rewards through a lower reputation score. Let $\delta \in (0, 1)$ be a discount factor. Deviation is unprofitable whenever

$$\delta \cdot \mathbb{E}[R_{\text{future}}(\mu_i)] > \text{short-term gain from deviation}. \tag{16}$$

Under this condition, no agent benefits from strategic misreporting, hence truthful submission is an equilibrium. □

Thus, verification combined with reputation dynamics implements a self-enforcing contract for knowledge quality.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

11.3 Reputation as Epistemic Capital

The trust score

$$\mu_i(t) = \frac{\sum_{K \in K_i^{\text{verified}}(t)} \text{conf}(K) \cdot \text{reuse}(K)}{\sum_{K \in K_i^{\text{total}}(t)} 1} \tag{17}$$

can be interpreted as *epistemic capital*.

Define discounted lifetime utility:

$$W_i = \sum_{t=0}^{\infty} \delta^t U_i(t), \qquad \delta \in (0,1). \tag{18}$$

Proposition 11.4 (Reputation Stability Condition)**.** *If*

$$\delta \cdot \frac{\partial\, \mathbb{E}[R_{\text{future}}]}{\partial \mu_i} > \textit{marginal short-term gain from deviation}, \tag{19}$$

then cooperative knowledge production is subgame perfect.

The inequality expresses that the discounted marginal benefit of preserving reputation exceeds any one-shot deviation incentive.

11.4 Threshold Parameter and Knowledge Growth Trade-off

The confidence threshold θ acts both as an epistemic filter and as a growth regulator. Define the knowledge growth rate:

$$g(\theta) = \frac{|\Sigma(t+1)| - |\Sigma(t)|}{|\Sigma(t)|}. \tag{20}$$

Proposition 11.5 (Threshold Trade-off)**.** *The threshold parameter satisfies:*

1. *if θ increases, then $g(\theta)$ decreases;*
2. *if θ decreases, then the average confidence $\bar{p}(\theta)$ decreases;*
3. *there exists $\theta^* \in (0,1)$ maximizing marginal collective intelligence growth:*

$$\theta^* = \arg\max_{\theta} \left(CI(t+1) - CI(t) \right). \tag{21}$$

Proof sketch. A higher threshold admits fewer derived units, reducing the increment of $|\Sigma(t)|$. A lower threshold admits more units but decreases mean confidence and may reduce the productivity factor in $CI(t)$. Existence of θ^* follows from standard continuity and compactness arguments on $(0,1)$. □

11.5 Collective Intelligence as Endogenous Knowledge Growth

Let

$$K(t) = |\Sigma(t)| \tag{22}$$

denote knowledge capital. By Proposition 11.5,

$$K(t) \geq K(0)\,(1+\lambda\gamma)^t. \tag{23}$$

Define effective cognitive output by

$$Y(t) = \alpha K(t)^{\beta}, \qquad \alpha > 0,\ \beta > 1. \tag{24}$$

Proposition 11.6 (Superlinear Cognitive Productivity)**.** *If $\beta > 1$, then marginal productivity is increasing in K:*

$$\frac{dY}{dK} = \alpha\beta K^{\beta-1}. \tag{25}$$

Hence, the system exhibits increasing returns to knowledge scale, consistent with emergent collective intelligence.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

11.6 Economic Interpretation of the Collective Intelligence Functional

Recall the definition

$$CI(t) = \frac{|\Sigma(t)|}{|\Sigma^*|} \cdot \bar{p}(t) \cdot \left(1 + \frac{C_s(t)}{T_s(t)}\right). \tag{26}$$

Define normalized knowledge capital

$$\kappa(t) = \frac{|\Sigma(t)|}{|\Sigma^*|}, \tag{27}$$

and denote $\rho(t) = C_s(t)/T_s(t)$. Then

$$CI(t) = \kappa(t) \cdot \bar{p}(t) \cdot (1 + \rho(t)). \tag{28}$$

Each factor admits an economic interpretation: $\kappa(t)$ corresponds to capital accumulation, $\bar{p}(t)$ to average quality, and $\rho(t)$ to collaborative productivity.

Proposition 11.7 (Monotone Growth of $CI(t)$)**.** *Under the assumptions of Theorem 4.4,*

$$CI(t+1) \geq CI(t). \tag{29}$$

Proof. By Theorem 4.4, $|\Sigma(t)|$ is non-decreasing and bounded above, while the threshold mechanism admits only knowledge units with $\mathrm{conf}(K) \geq \theta$, which prevents deterioration of mean confidence. The collaborative factor $C_s(t)/T_s(t)$ is non-decreasing as validated solutions accumulate. Therefore all multiplicative components of $CI(t)$ are non-decreasing, implying the claim. □

12 Discussion

The obtained results should be interpreted not only within the formal mathematical framework of the Distributed Knowledge Lattice, but also in the broader context of digital platform economics, cross-border logistics integration, digital twin architectures, and AI-based governance systems.

From the perspective of cross-border logistics system design, the integration of information, financial, and physical layers within the proposed framework corresponds to empirical evidence obtained in the context of AI-enhanced omnichannel logistics networks [36]. The present study extends this line of inquiry by providing a rigorous mathematical formalization of coordination effects, demonstrating that epistemic superadditivity in the lattice structure corresponds to measurable gains in system-level performance under adaptive routing and reinforcement learning.

At the level of global AI governance, the results align with the strategic perspective articulated in the framework of coordinated international AI development [37]. The formalization of incentive compatibility, reputation-based stabilization, and threshold optimization supports the idea that large-scale AI ecosystems require mathematically grounded mechanisms of trust, verification, and adaptive coordination. In this sense, the Distributed Knowledge Lattice may be interpreted as a micro-level formal analogue of institutional AI coordination structures proposed in the IAIC framework.

Thus, the scientific contribution of the present study lies in synthesizing:

- lattice-theoretic formalism of knowledge aggregation;
- incentive-compatible mechanism design;
- digital platform ecosystem theory;
- AI-enhanced logistics coordination;
- digital twin behavioral modeling.

By integrating these strands into a unified mathematical architecture, the study bridges abstract epistemic dynamics and applied economic system design, contributing to the theoretical foundation of AI-enabled cross-border intelligent infrastructures.

Relation to Prior Work. Our framework extends the task-based cognitive architecture of from individual agent cognition to population-level intelligence dynamics. While the TBCA demonstrated a closed cognitive loop for single agents with blockchain memory, the DKL formalism captures the emergent properties arising from *inter-agent* knowledge flows. Similarly, the logical-probabilistic learning theory of [12] is here elevated from a descriptive framework to a rigorous mathematical theory with convergence guarantees and complexity bounds.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1007 DOI: 10.66693/mathai.1007
Special Issue: MathAI 2026 Selected Papers
Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

The Non-Transitivity Problem. A fundamental insight driving our framework is that classical logical inference rules cannot be naively applied to probabilistic knowledge (Remark 3.8). This has profound implications for LLM-based reasoning, where models effectively operate with probabilistic data. The LP-Agent verification protocol addresses this by explicitly tracking confidence decay and triggering gap resolution when chains become too long.

Scalability Considerations. The multi-blockchain architecture provides horizontal scalability through chain partitioning. With N blockchain instances, write throughput scales linearly ($N \times 200$ tasks/sec) while read operations can query any instance without consensus overhead ($\sim$100 ms latency). The polynomial knowledge retrieval algorithm (Theorem 7.1) ensures that the DKL remains efficiently queryable as collective knowledge grows.

Limitations and Future Directions. Current limitations include: (i) the formalization of commonsense reasoning remains challenging for LP-Agents; (ii) the confidence threshold θ requires domain-specific tuning; (iii) adversarial agents may attempt to inject misleading knowledge, requiring more sophisticated Byzantine fault tolerance in the epistemic governance layer. Future work will explore: integration of CRDTs for eventual consistency in knowledge sharing, meta-cognitive modules that dynamically decide when to engage full verification versus heuristic reasoning, and cross-domain knowledge transfer across different Enigma metaverse instances.

13 Conclusion

This paper has presented a rigorous mathematical framework for the emergence of CI in metaverse ecosystems. By introducing the Distributed Knowledge Lattice as a formal substrate, we have proven that under precisely stated conditions, multi-agent knowledge dynamics converge to a fixed point representing an emergent cognitive state greater than any individual agent's capabilities (Theorem 4.4). The superadditivity result (Theorem 7.3) formally establishes the intuition that CI is more than the sum of its parts.

The LP-Agent verification framework provides the trustworthiness and explainability guarantees essential for deploying CI in high-stakes applications, while the multi-blockchain infrastructure ensures that collective knowledge is persistent, tamper-proof, and economically incentivized. The integration of LLMs as hypothesis generators within a formal verification pipeline addresses the fundamental limitations of contemporary AI systems—hallucination, static knowledge, and opacity—while preserving their creative generative capabilities.

The Enigma Metaverse thus represents a paradigm shift: from AI systems that merely coexist in a shared virtual space to a *cognitive ecosystem* where intelligence itself is a collective, evolving, verifiable, and self-optimizing property. This work provides the mathematical foundations for realizing this vision and opens new research directions at the intersection of mathematical logic, probability theory, distributed systems, and artificial intelligence.

Acknowledgments

This work was supported by a grant for research centers, provided by the Ministry of Economic Development of the Russian Federation in accordance with the subsidy agreement with Novosibirsk State University. The authors thank the Sobolev Institute of Mathematics and the International Artificial Intelligence Committee (IAIC) for their support.

References

[1] T. Brown, B. Mann, N. Ryder, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020. https://arxiv.org/abs/2005.14165

[2] V. Buterin. A next-generation smart contract and decentralized application platform. *Ethereum White Paper*, 2014. https://ethereum.org/en/whitepaper/

[3] L. De Raedt, K. Kersting, S. Natarajan, and D. Poole. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. Morgan & Claypool, 2016. https://doi.org/10.2200/S00692ED1V01Y201601AIM032

[4] L. Gao, A. Madaan, S. Zhou, et al. PAL: Program-aided language models. In *Proc. ICML*, PMLR 202:10764–10799, 2023. https://arxiv.org/abs/2211.10435

[5] L. Getoor and B. Taskar. *Introduction to Statistical Relational Learning*. MIT Press, 2007. https://mitpress.mit.edu/9780262538688/

[6] S. Goncharov and A. Nechesov. Axiomatization of blockchain theory. *Mathematics*, 11(13):2966, 2023. https://doi.org/10.3390/math11132966

[7] M. Hämäläinen. Urban development with dynamic digital twins in Helsinki city. *IET Smart Cities*, 3(4):201–210, 2021. https://doi.org/10.1049/smc2.12015

[8] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the AMS*, 43(4):439–561, 2006. https://doi.org/10.1090/S0273-0979-06-01126-8

[9] C. Udokwu, R. Voicu-Dorobanţu, A. A. Ogunyemi, A. Norta, N. Sturua, and S. Craß. Leveraging blockchain for ethical AI: Mitigating digital threats and strengthening societal resilience. *Future Internet*, 17:309, 2025. https://doi.org/10.3390/fi17070309

[10] L.-H. Lee, T. Braud, P. Zhou, et al. All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda. *arXiv preprint arXiv:2110.05352*, 2021. https://arxiv.org/abs/2110.05352

[11] G. Li, H. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in NeurIPS*, 2023. https://arxiv.org/abs/2303.17760

[12] A. Nechesov, I. Dorokhov, and J. Ruponen. Virtual cities: From digital twins to autonomous AI societies. *IEEE Access*, 13:13866–13903, 2025. https://doi.org/10.1109/ACCESS.2025.3531222

[13] L. Pan, A. Albalak, X. Wang, and W. Y. Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of EMNLP*, 2023. https://arxiv.org/abs/2305.12295

[14] M. S. Paterson and M. N. Wegman. Linear unification. *Journal of Computer and System Sciences*, 16(2):158–167, 1978. https://doi.org/10.1016/0022-0000(78)90043-0

[15] R. Riegel, A. Gray, F. Luus, et al. Logical neural networks. *arXiv preprint arXiv:2006.13155*, 2020. https://arxiv.org/abs/2006.13155

[16] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 4th edition, 2022. https://aima.cs.berkeley.edu/

[17] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009. https://www.masfoundations.org/

[18] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee. Digital twin in industry: State-of-the-art. *IEEE Trans. Ind. Informat.*, 15(4):2405–2415, 2019. https://doi.org/10.1109/TII.2018.2873186

[19] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, 1955. https://projecteuclid.org/journals/pacific-journal-of-mathematics/volume-5/issue-2/A-lattice-theoretical-fixpoint-theorem-and-its-applications/pjm/1103044538.full

[20] H. Touvron, L. Martin, K. Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. https://arxiv.org/abs/2307.09288

[21] M. Wooldridge. *An Introduction to MultiAgent Systems*. Wiley, 2nd edition, 2009. https://www.cs.ox.ac.uk/people/michael.wooldridge/pubs/imas/IMAS2e.html

[22] Altera.AI. et al. Project Sid: Many-agent simulations toward AI civilization. *arXiv preprint arXiv:2411.00114*, 2024. https://arxiv.org/abs/2411.00114

[23] S. Goncharov and A. Nechesov. Polynomial-computable representation of neural networks in semantic programming. *J*, 6(1):48–57, 2023. https://doi.org/10.3390/j6010004

[24] Y. LeCun. A path towards autonomous machine intelligence. *OpenReview*, 2022. https://openreview.net/forum?id=BZ5a1r-kVsf

[25] H. Liu et al. Trustworthy AI: A computational perspective. *ACM Trans. Intell. Syst. Technol.*, 14(1):4, 2022. https://doi.org/10.1145/3546872

Received: 03 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

[26] N. F. Rajani et al. Explain yourself! Leveraging language models for commonsense reasoning. ArXiv. In *Proc. ACL*, 2019. https://arxiv.org/abs/1906.02361

[27] A. S. Abadi and L.-K. Soh. Challenges in credit assignment for multi-agent reinforcement learning in open agent systems. *arXiv preprint arXiv:2510.27659*, 2025. https://arxiv.org/abs/2510.27659

[28] W. Chen, Y. Su, J. Zuo, C. Yang, C. Yuan, et al. AgentVerse: Facilitating multi-agent collaboration and exploring emergent behaviors. *arXiv preprint arXiv:2308.10848*, 2023. https://arxiv.org/abs/2308.10848

[29] W. Chen, W. Li, X. Liu, S. Yang, and Y. Gao. Learning explicit credit assignment for cooperative multi-agent reinforcement learning via polarization policy gradient. *arXiv preprint arXiv:2210.05367*, 2023. https://arxiv.org/abs/2210.05367

[30] S. Han, Q. Zhang, W. Jin, and Z. Xu. LLM multi-agent systems: Challenges and open problems. *arXiv preprint arXiv:2402.03578*, 2024. https://arxiv.org/abs/2402.03578

[31] Z. Ning and L. Xie. A survey on multi-agent reinforcement learning and its application. *Journal of Automation and Intelligence*, 3(2):73–91, 2024. https://doi.org/10.1016/j.jai.2024.02.003

[32] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023. https://arxiv.org/abs/2304.03442

[33] K.-T. Tran, D. Dao, M.-D. Nguyen, Q.-V. Pham, B. O'Sullivan, and H. D. Nguyen. Multi-agent collaboration mechanisms: A survey of LLMs. *arXiv preprint arXiv:2501.06322*, 2025. https://arxiv.org/abs/2501.06322

[34] S. Gronauer and K. Diepold. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2):895–943, 2022. https://doi.org/10.1007/s10462-021-09996-w

[35] S. Wu and K. Shu. Memory in LLM-based multi-agent systems: Mechanisms, challenges, and collective intelligence. 2025. https://doi.org/10.13140/RG.2.2.21084.04485

[36] S. Barykin, W. Zhang, D. Dinets, A. Nechesov, et al. Designing a Russian–Chinese omnichannel logistics network for the supply of bioethanol. *Sustainability*, 17:7968, 2025. https://doi.org/10.3390/su17177968

[37] A. Nechesov and S. Barykin. One AI, One World: A global AI strategy by the International AI Committee (IAIC). *System Informatics*, 26:77–102, 2025. https://doi.org/10.31144/si.2307-6410.2025.n26.p77-102

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1009 DOI: 10.66693/mathai.1009
Special Issue: MathAI 2026 Selected Papers
Received: 05 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

OLoRA+: A Hybrid Approach to Parameter-Efficient Fine-Tuning of Large Language Models

S.I.Kushmuratov, F.T. Adilova, R.R. Davronov
Laboratory of Biomedical Informatics
V.I.Romanovskiy Institute of Mathematics, Uzbekistan Academy of Sciences
9 University Street, Tashkent 100174, Uzbekistan
bekmezonali@gmail.com, fatadilova@mathinst.uz, rifqat@gmail.com

Abstract

Parameter-Efficient Fine-Tuning (PEFT) is essential for adapting Large Language Models (LLMs) under resource constraints, yet existing methods often treat initialization and optimization as separate concerns. This paper introduces OLoRA+, a novel hybrid approach that synergistically combines the structural stability of Orthonormal Low-Rank Adaptation (OLoRA) with the accelerated convergence of LoRA+. By initializing adapter matrices via QR decomposition of pre-trained weights and applying differential learning rates to the upstream and downstream projection matrices, OLoRA+ aims to enhance both stability and feature learning speed. We evaluated the method on the LLMs models using a subset of the Alpaca instruction-following dataset. Empirical results demonstrate that OLoRA+ consistently outperforms the standard OLoRA baseline across Evaluation Loss, BLEU, and ROUGE metrics without incurring additional computational costs. Crucially important that our analysis uncovers two distinct effective learning regimes: a "Refinement" strategy (learning rate ratio $\lambda < 1$) that optimizes the initial orthonormal basis, and an "Exploration" strategy ($\lambda > 1$) that seeks new parameter directions. These findings suggest that OLoRA+ offers a more versatile and robust framework for efficient LLM adaptation than its predecessors.

Keywords: LLM, Fine-tuning, PEFT, Low-Rank Adaptation, QR Decomposition, Differential Optimization.

1 Introduction

The advent of pre-trained large language models (LLMs) has significantly advanced the field of natural language processing (NLP) (Bommasani et al., 2021). These models demonstrate powerful capabilities in general language understanding and generation. However, the increasing size of LLMs presents significant challenges for traditional full fine-tuning, which updates all model parameters and incurs substantial computational and memory costs (Devlin et al., 2018).

In response to these challenges, Parameter-Efficient Fine-Tuning (PEFT) methods have emerged as a promising solution. PEFT techniques adapt pre-trained models by selectively fine-tuning a small number of additional parameters while keeping the majority of the original model frozen. Among these, Low-Rank Adaptation (LoRA) (Hu et al., 2021) has become one of the most popular approaches, achieving strong performance without increasing inference latency. The success of LoRA has spurred the development of an ecosystem of variants, each aimed at addressing specific limitations of the original method.

Research into improving LoRA has progressed along several distinct directions. On one hand, methods like OLoRA have focused on the initialization of the adapter (Büyükakyüz, 2024). OLoRA utilizes QR decomposition to create an orthonormal basis from the pre-trained weights, providing a more stable and well-conditioned starting point for training, which leads to accelerated convergence. On the other hand, approaches like LoRA+ have focused on the optimization of the training process.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1009 DOI: 10.66693/mathai.1009
Special Issue: MathAI 2026 Selected Papers
Received: 05 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

LoRA+ demonstrates that using differential learning rates for the two adapter matrices (A and B) can significantly accelerate training and improve final performance (Hayou et al., 2024).

However, these research directions—improving initialization and optimizing the training process—have largely evolved in parallel. The potential synergistic effects of their combination remain an unexplored area. This paper addresses this gap by introducing OLoRA+, a novel hybrid method that synergistically combines the structured initialization of OLoRA with the accelerated optimization of LoRA+.

The primary goal of this research is to investigate the interaction between these two techniques and determine if their combination yields a more effective and versatile PEFT method. Through empirical evaluation, we not only compare the performance of OLoRA+ against the OLoRA baseline but also uncover a new learning dynamic related to the choice of the learning rate ratio. We characterize these regimes as "Refinement" and "Exploration" strategies, demonstrating that OLoRA+ surpasses its predecessors and offers a deeper understanding of the interplay between adapter initialization and optimization.

2 Related Work

Our work builds directly upon three foundational PEFT methods: LoRA (Hu et al., 2021), OLoRA (Büyükakyüz, 2024), and LoRA+ (Hayou et al., 2024).

2.1 Low-Rank Adaptation (LoRA)

LoRA is based on the hypothesis that the change in a model's weights during adaptation occurs in a "low-dimensional space". It freezes the pre-trained model weights and injects a pair of trainable low-rank matrices, A and B, into each target layer which is shown in Figure 1. The update is represented as $\Delta W = BA$.

$$W = W_0 + \Delta W = W_0 + BA \tag{1}$$

To ensure a non-disruptive start to training, matrix B is initialized to zeros, making the initial update zero. While highly parameter-efficient, LoRA can converge more slowly than full fine-tuning (Hu et al., 2021).

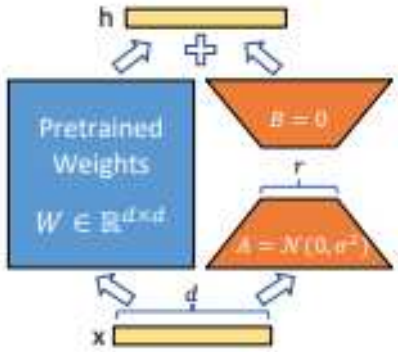


Figure 1: Reparametrization of weights for train only A and B

2.2 Orthonormal LoRA (OLoRA)

OLoRA addresses the convergence speed and stability limitations of LoRA by improving the initialization process. Instead of random initialization, OLoRA performs a QR decomposition on the pre-trained weight matrix W to obtain an orthogonal matrix Q and an upper-triangular matrix R (Büyükakyüz, 2024). The adapter matrices B and A are then initialized with the first r columns of

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1009 DOI: 10.66693/mathai.1009
Special Issue: MathAI 2026 Selected Papers
Received: 05 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Q and r rows of R (Aghajanyan et al., 2020; Meng et al., 2024), respectively, which is shown in Figure 2.

$$W_{\text{adapted}} = W + Q_r R_r \tag{2}$$

This orthonormal initialization provides a "well-conditioned optimization landscape," leading to faster convergence and often better final performance compared to standard LoRA.

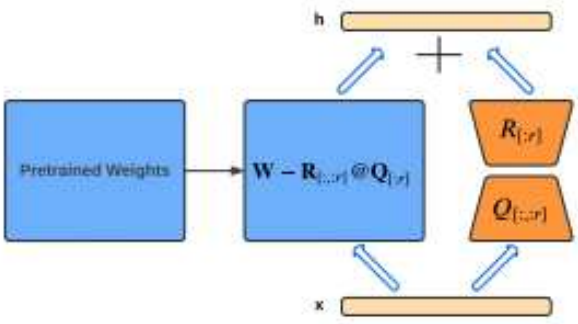


Figure 2: Illustration of the OLoRA method

2.3 LoRA+

LoRA+ identifies a suboptimality in the standard LoRA training procedure, where both adapter matrices A and B are updated with the same learning rate. The authors (Hayou et al., 2024) argue that this slows down feature learning. LoRA+ corrects this by setting a much higher learning rate for matrix B than for matrix A (i.e., $\eta_B = \lambda \eta_A$, where $\lambda \gg 1$). This simple change is shown to improve performance by 1-2% and accelerate training speed by up to 2x at no additional computational cost.

$$A \leftarrow A - \eta \times G_A, \quad B \leftarrow B - \lambda\eta \times G_B, \quad \lambda \gg 1 \tag{3}$$

3 The OLoRA+ Method

Our central hypothesis is that the benefits of OLoRA's superior initialization and LoRA+'s accelerated optimization are complementary and can be combined to create a more powerful PEFT method.

3.1 Architectural Synergy

OLoRA+ is a hybrid method that integrates these two techniques in a two-phase conceptual process: a structured initialization followed by a differential optimization.

The process begins with the **OLoRA initialization phase**. For a given pre-trained weight matrix W, we first perform a QR decomposition, $W = QR$. The adapter matrices, denoted as B (up-projection) and A (down-projection), are then initialized using the results of this decomposition. Specifically, B is initialized with Q_r (the first r columns of the orthogonal matrix Q) and A is initialized with R_r (the first r rows of the upper-triangular matrix R). This provides a structured, orthonormal starting point that is a low-rank approximation of the original weights, creating a "well-conditioned optimization landscape" (Büyükakyüz, 2024). The adapted weight matrix is thus expressed as $W_{\text{adapted}} = W + BA$, where B and A start as Q_r and R_r.

$$W = W_0 + \Delta W = W_0 + BA = W_0 + Q_r R_r \tag{4}$$

Once training begins, the **LoRA+ optimization phase** is applied. Instead of using a single learning rate η for both adapter matrices, we introduce a learning rate ratio, λ. The update rules for the matrices during each step of gradient descent are governed by the following formulas:

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1009 DOI: 10.66693/mathai.1009
Special Issue: MathAI 2026 Selected Papers
Received: 05 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

$$A \leftarrow A - \eta \times G_A \tag{5}$$
$$B \leftarrow B - \lambda\eta \times G_B \tag{6}$$

Here, G_A and G_B represent the gradients for matrices A and B, respectively. The base learning rate is η, and the effective learning rate for matrix B is scaled by the ratio λ. The LoRA+ paper recommends $\lambda \gg 1$ for standard LoRA to accelerate feature learning. This synergy—starting from a superior initial state and then following a more efficient learning path—is the core of the OLoRA+ method.

3.2 Implementation Details

We implemented OLoRA+ by leveraging the Hugging Face PEFT library.

- **OLoRA Initialization:** We configured the `LoraConfig` by setting the `init_lora_weights="olora"` parameter. This instructs the library to perform the QR decomposition and initialize the A and B matrices accordingly.
- **LoRA+ Optimization:** We created a custom optimizer by separating the model's trainable parameters into two groups based on their names (A and B). We then instantiated an AdamW optimizer, assigning different learning rates to each group based on a specified λ. This custom optimizer was then passed to the Hugging Face Trainer.

4 Experimental Setup

To validate our method, we conducted a controlled experiment comparing OLoRA+ against a standard OLoRA (Büyükakyüz, 2024) baseline.

4.1 Models

All experiments were conducted using the TinyLlama-1.1B, OPT-1.3B and Gemma-2B models, compact yet capable LLMs suitable for efficient experimentation (Zhang et al., 2024).

4.2 Dataset

We used a subset of the tatsu-lab/alpaca dataset, which consists of 52,000 instruction-following demonstrations generated by OpenAI's text-davinci-003 engine (Taori et al., 2023). For our experiments, we used a split of 1000 samples for training and 500 samples for evaluation to simulate a resource-constrained fine-tuning scenario.

4.3 Baselines and Hyperparameters

To ensure a fair comparison, all key hyperparameters were kept consistent across experiments:

- **Rank (r):** 32
- **Alpha (α):** 16
- **Learning Rate:** 3e-4
- **Epochs:** 1
- **Baseline (OLoRA):** The model was trained using the OLoRA initialization with a standard AdamW optimizer (Loshchilov & Hutter, 2017).
- **Our Method (OLoRA+):** The model was trained using the OLoRA initialization and our custom LoRA+ optimizer with a varying learning rate ratio.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1009 DOI: 10.66693/mathai.1009
Special Issue: MathAI 2026 Selected Papers
Received: 05 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

4.4 Evaluation Metrics

Model performance was assessed using a suite of standard metrics:

- **Evaluation Loss:** To measure the model's ability to generalize to unseen data during training (Wikipedia contributors, 2025b).
- **BLEU:** To measure the precision and fluency of the generated text (Wikipedia contributors, 2025a).
- **ROUGE:** Specifically ROUGE-1, ROUGE-2, and ROUGE-L, to measure the recall of key information in the generated text (Wikipedia contributors, 2025c).

5 Results and Discussion

Our experiments confirmed our initial hypothesis and led to a significant new discovery regarding the learning dynamics of initialized adapters.

5.1 Performance Improvement

Across all quantitative metrics, our OLoRA+ method consistently outperformed the OLoRA baseline. The model trained with OLoRA+ achieved a lower final evaluation loss and higher BLEU and ROUGE scores, indicating superior generalization and text generation quality which is shown in Tables 1, 2, and 3.

Table 1: Evaluation results by TinyLlama-1.1B

Method	Eval Loss	ROUGE-1	ROUGE-2	ROUGE-L	BLEU
OLoRA (Baseline)	88.49	74.53	56.46	71.78	56.81
OLoRA+ ($\lambda \ll 1$)	88.22	74.56	56.41	71.86	56.83
OLoRA+ ($\lambda \gg 1$)	88.39	74.52	56.49	71.86	56.83

Table 2: Evaluation results by OPT-1.3B

Method	Eval Loss	ROUGE-1	ROUGE-2	ROUGE-L	BLEU
OLoRA (Baseline)	75.72	63.53	62.32	73.08	68.45
OLoRA+ ($\lambda \ll 1$)	75.68	62.56	62.38	73.13	68.56
OLoRA+ ($\lambda \gg 1$)	75.41	64.52	62.33	73.25	68.53

Table 3: Evaluation results by Gemma-2B

Method	Eval Loss	ROUGE-1	ROUGE-2	ROUGE-L	BLEU
OLoRA (Baseline)	81.22	83.78	59.25	78.56	62.14
OLoRA+ ($\lambda \ll 1$)	81.18	83.82	59.36	78.59	62.19
OLoRA+ ($\lambda \gg 1$)	81.13	83.81	59.33	78.61	62.19

As shown in Figure 3 and Table 4, our proposed OLoRA+ method demonstrates clear superiority across all metrics. According to (a), Tiny-llama demonstrated significantly lower testing losses than the other LLMs. Graphs (b), (c), (d), and (e) show the performance for the BLEU and ROUGE text generation metrics. In each case, both OPT-1.3 and Gemma-2B LLM outperform or equal the baseline.

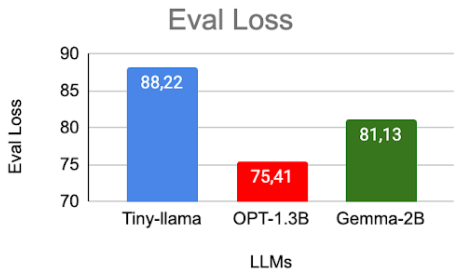
5.2 The Discovery of Dual Learning Regimes

The most significant finding of our research is that, contrary to the recommendations in the original LoRA+ paper, OLoRA+ achieves strong performance with a learning rate ratio (λ) both greater

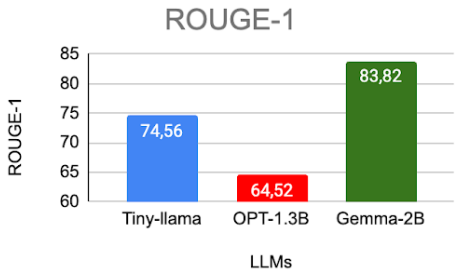
Mathematics & AI, Vol. 1, No. 2, 2026, Article 1009 DOI: 10.66693/mathai.1009
Special Issue: MathAI 2026 Selected Papers
Received: 05 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Table 4: Evaluation results by LLMs

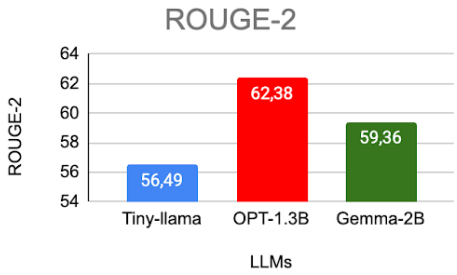
LLMs	Eval Loss	ROUGE-1	ROUGE-2	ROUGE-L	BLEU
Tiny-llama	88.22	74.56	56.49	71.86	56.83
OPT-1.3B	75.41	64.52	62.38	73.25	68.56
Gemma-2B	81.13	83.82	59.36	78.61	62.19



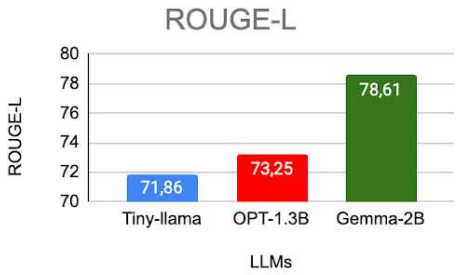
(a) Evaluation Loss



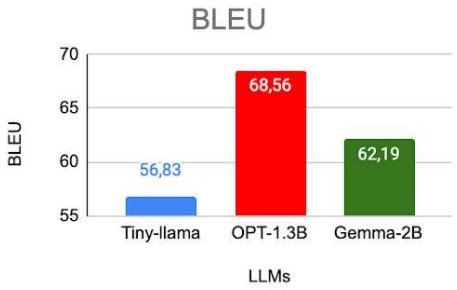
(b) ROUGE-1 score



(c) ROUGE-2 score



(d) ROUGE-L score



(e) BLEU score

Figure 3: Comparative analysis of the performance of metrics by LLMs.

than 1 and less than 1. This discovery suggests that the optimal optimization strategy is contingent on the initialization method. We characterize these two effective regimes as a trade-off between ”Refinement” and ”Exploration.”

5.3 Characterizing ”Refinement” vs. ”Exploration”

The Refinement Strategy ($\lambda < 1$): When the learning rate for matrix A (derived from R_r) is higher than for matrix B (derived from Q_r), the model adopts a conservative strategy. It trusts the high-quality orthonormal directions provided by the OLoRA initialization and focuses the learning

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1009 DOI: 10.66693/mathai.1009
Special Issue: MathAI 2026 Selected Papers
Received: 05 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

effort on finding the correct magnitudes and combinations for these directions. This approach gently refines the strong initial structure. Our results show this is a highly effective strategy, suggesting that for many tasks, the primary challenge is learning *how* to use the initial basis, not finding a new one.

The Exploration Strategy ($\lambda > 1$): When the learning rate for matrix B is higher than for matrix A, the model adopts a more aggressive strategy. It uses the OLoRA initialization as a launchpad but actively explores for new directions in the parameter space by more rapidly updating the B matrix. This approach is more willing to deviate from the initial orthonormal structure in search of a potentially better solution. This aligns with the original LoRA+ logic and is effective for tasks that may require adaptation beyond the initial subspace.

6 Conclusion

In this paper, we introduced OLoRA+, a novel hybrid PEFT method that synergistically combines the orthonormal initialization of OLoRA with the differential optimization of LoRA+. Results of computational experiments demonstrate that OLoRA+ consistently outperforms the OLoRA baseline in instruction-following tasks, achieving lower evaluation loss and higher BLEU and ROUGE scores via different LLMs which are TinyLlama-1.1B, OPT-1.3B and Gemma-2B without any additional computational cost.

The key novelty of this study is the discovery and characterization of two distinct and effective learning regimes for OLoRA+: a "Refinement" strategy (ratio < 1) and an "Exploration" strategy (ratio > 1). This finding reveals that the optimal optimization strategy is fundamentally dependent on the initialization scheme, offering a new dimension for hyperparameter tuning. This research not only presents OLoRA+ as a more versatile and powerful approach for LLM adaptation but also provides a deeper understanding of the critical interplay between adapter initialization and optimization dynamics, paving the way for more effective fine-tuning in resource-constrained environments.

References

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

Rishi Bommasani et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Kaan Büyükakyüz. Olora: Orthonormal low-rank adaptation of large language models. *arXiv preprint arXiv:2406.01775*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024.

Edward J Hu et al. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Fanxia Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. *arXiv preprint arXiv:2404.02948*, 2024.

Rohan Taori et al. Stanford alpaca: An instruction-following llama model, 2023.

Wikipedia contributors. Bleu, 2025a. URL `https://en.wikipedia.org/wiki/BLEU`.

Wikipedia contributors. Evaluation function, 2025b. URL `https://en.wikipedia.org/wiki/Evaluation_function`.

Wikipedia contributors. Rouge (metric), 2025c. URL `https://en.wikipedia.org/wiki/ROUGE_(metric)`.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1009 DOI: 10.66693/mathai.1009
Special Issue: MathAI 2026 Selected Papers
Received: 05 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

INTERPRETABLE EMOTION ATTRIBUTION IN SOCIAL GRAPHS: A COMPARATIVE ANALYSIS OF RULE-BASED, TRANSFORMER, AND LLM MODELS

Usman Gidado & Anton Kolonin
Department of Mathematics & Mechanics
Novosibirsk State University
Novosibirsk, Russia
{ubgidadoac,akolonin}@gmail.com

ABSTRACT

Emotion attribution in social graphs requires inferring directed emotional attitudes between entities in complex, multi-turn dialogues. While transformer models dominate the field, they often lack the transparency required for social science applications. We present a systematic comparison of three modeling paradigms for this task: a fully interpretable rule-based system, a fine-tuned RoBERTa-large model, and a few-shot Llama-3-8B. Utilizing the DialogRE dataset, we demonstrate that incorporating a 3-turn conversational context significantly improves attribution accuracy across paradigms. Crucially, our results show that the interpretable rule-based system achieves a competitive F1 score, making it statistically indistinguishable from the state-of-the-art RoBERTa model. In contrast, we find that few-shot large language models exhibit poor performance in emotion attribution of semantic relations, performing below the rule-based baseline. We prove that interpretability does not necessitate a performance trade-off in social emotion analysis.[1] [2]

1 INTRODUCTION

Emotion Recognition in Conversation (ERC) has primarily focused on identifying the emotional state expressed by an individual speaker at a given time (Khalid & Sano, 2023). While this formulation is effective for monologic or dyadic sentiment analysis, it is insufficient for complex multi-party interactions, where emotions are often directed toward specific entities rather than expressed in isolation. In such settings, understanding who feels what toward whom becomes essential. This problem, commonly referred to as emotion attribution, naturally induces a social graph representation, where the entities form nodes and the directed edges encode emotional relations between them (Dekker et al., 2019; Khemani et al., 2024).

Emotion attribution over social graphs is increasingly relevant in real-world applications which include social media analysis, organizational communication, mental health monitoring, and legal or forensic dialogue interpretation. In these domains, predictive accuracy alone is not sufficient: systems must also support interpretability, allowing practitioners to inspect, validate, and correct model decisions when necessary (Tan et al., 2025; Sarma, 2019). This requirement is particularly acute in regulated or high-stakes environments, where opaque decision-making can hinder trust, accountability, and deployment. Despite these constraints, the dominant approaches to emotion attribution in recent NLP research rely on black-box neural architectures, most notably fine-tuned Transformer encoders and, more recently, large language models (LLMs) used in few-shot or zero-shot settings (Lalk et al., 2025; Mohammadi et al., 2025). While these models often achieve strong empirical performance, they provide limited insight into why a particular emotional relation is predicted, and inability to trace its decision back to a specific linguistic trigger, this prevents practitioners from performing systematic error correction. In contrast, rule-based approaches, which offer full

[1] **Code:** https://github.com/UBGidado/Interpretable-EA-in-SG
[2] **Model:** https://github.com/aigents/pygents/tree/main/pygents

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

transparency and explicit reasoning, are often dismissed as outdated or fundamentally less capable, particularly in comparison to large-scale transformer models.

In this work, we revisit this assumption through a comparative study of emotion attribution in social graphs, explicitly examining the trade-offs between performance and interpretability across rule-based, Transformer, and few-shot LLM approaches under a unified relational representation. While LLMs demonstrate remarkable zero-shot capabilities, it remains unclear whether few-shot prompting can match the calibrated decision boundaries of smaller, task-specific fine-tuned models in nuanced domains. This question is particularly relevant given the growing reliance on in-context learning. However, systematic experiments with Llama-3-8B-Instruct yield significantly lower performance, achieving a maximum Macro-F1 of 0.528 (10 few-shot) and 0.501 in our prompt ablation study. A systematic ablation across multiple prompt templates and shot counts produces F1 scores ranging from 0.294 to 0.501, indicating that prompt engineering alone cannot close the gap with fine-tuned models. In contrast, our interpretable rule-based system achieves performance statistically comparable to fine-tuned Transformers, challenging the assumption that interpretability necessarily requires sacrificing accuracy. However, our evaluation is limited to DialogRE; generalization to other dialogue domains requires further study.

Our work makes three key contributions to interpretable emotion attribution in social graph:

1. We demonstrate that interpretable rule-based approaches achieve statistically competitive performance with fine-tuned transformers (see Table 2), challenging the assumption that explainability necessarily sacrifices substantial accuracy. This finding has important implications for deploying emotion attribution systems in regulated domains requiring transparency.
2. We show that few-shot prompting of large language models (Llama-3-8B) significantly underperforms targeted fine-tuning (see Table 2), even with sophisticated prompt engineering. This counter-intuitive result demonstrates that task-specific adaptation remains essential despite recent advances in foundation model scale and capabilities.
3. We provide the systematic comparison of explainability–performance trade-offs across three paradigms; rule-based, neural, and LLM approaches with statistical validation. This enables informed decisions about when interpretability can be prioritized without sacrificing performance.

2 Related Work

2.1 Interpretability in NLP

Interpretability in NLP is divided into post-hoc explanation and inherent transparency. Post-hoc methods, such as LIME and SHAP (Ribeiro et al., 2016; Lundberg & Lee, 2017), approximate the behavior of black-box models by estimating feature importance. Despite their widespread adoption, these approaches are often criticized for their dependence on the underlying machine learning model and their vulnerability to feature collinearity, underscoring the need for caution in their application and interpretation to prevent misleading analyses. (Salih et al., 2024). In contrast, inherently interpretable models are transparent by design. This category includes rule-based systems, decision trees, and weighted n-gram models. The Aigents framework exemplifies this paradigm by employing weighted structured patterns and heterarchical n-grams for multi-factor classification (Kolonin & Arinicheva, 2025), enabling explicit inspection and modification of decision logic. Such transparency is particularly valuable in domain-specific or high-stakes settings, where bias correction and rule-level validation are required (Salih et al., 2024).

More recently, mechanistic interpretability has emerged as an effort to reverse-engineer neural models by identifying neurons or circuits associated with linguistic concepts (Rai et al., 2024). These methods are promising, but remain computationally intensive and difficult to audit without specialized expertise (Sharkey et al., 2025). This tension between the generalization strength of neural models and the auditability of symbolic systems has driven the development of hybrid approaches that integrate neural pattern recognition with rule-based validation (Kolli, 2025).

Interpretable NLP models, such as rule-based or symbolic systems, have long been valued for transparency, especially in high-stakes tasks (Kolonin & Arinicheva, 2025). For example, recent work

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

by Lalk et al. (2025) applies step-by-step reasoning in sentiment analysis. However, such explanations often trade factual accuracy for interpretability. Our rule-based Aigents system follows this paradigm. Unlike prior ERC work that focuses on improving accuracy with complex neural architectures (Kim & Vossen, 2021), we explicitly examine how a fully interpretable system compares to opaque models, quantifying the interpretability–performance trade-off in relational emotion attribution. This allows for a nuanced understanding of model behavior, crucial for domains where accountability and trustworthiness are paramount (Tak et al., 2025). For instance, directed acyclic graph networks and graph neural networks have demonstrated improved performance in modeling conversational context for emotion recognition, yet their interpretability often remains limited (Khemani et al., 2024; Shen et al., 2021).

2.2 LLM Data Augmentation

Large Language Models have recently been leveraged to generate or label additional data in low resource settings like ours (Zhang et al., 2023b). GPT-based models can synthesize training examples by prompt-based few-shot generation (Kaddour & Liu, 2023), and have been used to create dialogue or summary data for related tasks. Feng et al. (2021) use DialoGPT to annotate summarization features in conversational text. And more generally, Zhang et al. (2023a) introduce an active learning framework (LLMaAA) in which an LLM acts as an annotator to label unlabeled inputs, improving classification with minimal human cost. Such LLM-augmented data has been shown to boost performance on complex tasks. In our work, we use few-shot LLaMA-3 prompts to generate emotion-label examples from our training set, following this line of research. We compare model performance with and without LLM-augmented examples, studying how data augmentation interacts with interpretability.

2.3 Social Graph Emotion Attribution

Recent work has begun to incorporate relational and graph structure into emotion prediction. Yu et al. (2020) introduced the DialogRE dataset for dialogue-based relation extraction, emphasizing speaker-aware modeling to capture cross-utterance relations . Graph-based ERC models extend this idea: for example, Liu et al. (2023) build heterogeneous conversation graphs to propagate contextual cues. These approaches improve emotion recognition in multi-party dialogues by explicitly modeling interactions. Similarly, in social media contexts, researchers have applied GNNs to predict users' emotions from text and network structure.For instance, an enhanced Text-GCN model employing a Pointwise Mutual Information (PMI)–based graph construction strategy achieves high emotion classification performance (78.64% and 92.38% accuracy) on Twitter datasets, while incorporating interpretability components such as attention-based word importance (Khemani et al., 2025). These explanations remain primarily post-hoc and model-internal. They do not provide auditable reasoning over relational emotional attributions between entities. In contrast, our work focuses on inherent interpretability at the level of social relations, enabling direct inspection and modification of emotion attribution logic across graph edges. Furthermore, our study is novel in comparing interpretable (rule-based) and black-box (transformer or LLM) approaches on this graph-based emotion attribution task. This allows us to assess how model design affects both performance and explainability in a relational emotion setting.

3 Problem Formulation

We formulate emotion attribution as a relation extraction task over multi-turn dialogues. Given a dialogue and an entity pair (subject, object), the goal is to classify the emotional relationship from subject to object as positive, negative, or neutral based on conversational context.

3.1 Task Definition

Following the dialogue-based relation extraction framework of Yu et al. (2020), we define our task formally as follows. Given a dialogue $D = \{(s_1, t_1), (s_2, t_2), \ldots, (s_m, t_m)\}$ where s_i denotes the speaker identifier and t_i represents the utterance text of the i-th turn, and an entity pair $(e_{\text{subj}}, e_{\text{obj}})$, the task is to predict the relation type $r \in \mathcal{R}$ from e_{subj} to e_{obj} based on D, where $\mathcal{R} = \{\text{positive}, \text{negative}, \text{neutral}\}$.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Formally, the objective is to learn a function

$$f : (D, e_{\text{subj}}, e_{\text{obj}}) \to \mathcal{R},$$

which maps a dialogue and an ordered entity pair to an emotion label.

Entity Representation: We represent entity pairs using special marker tokens. For an entity pair $(e_{\text{subj}}, e_{\text{obj}})$, we surround each entity mention in the dialogue with special tokens: `[SS]` e_{subj} `[/SS]` for the subject and `[OS]` e_{obj} `[/OS]` for the object. This enables models to attend to relevant entity mentions throughout the conversation. For speaker entities, we mark the speaker identifier at the beginning of each turn. The concatenated dialogue with entity markers serves as input to relation extraction models.

3.3 Dataset

We utilize DialogRE v2 (Yu et al., 2020), a first human-annotated dialogue-based relation extraction dataset derived from *Friends* TV show transcripts, consisting of 1,073 training, 358 development, and 357 test dialogues. The dataset contains 10,168 relation instances spanning 36 relation types, which we organize into five high-level categories: Identity & Metadata (45.0%), including alternate names and unanswerable cases; Personal Relationships (16.1%), such as friends (7.1%) and siblings (3.1%); Interpersonal Sentiment via positive (7.6%) and negative (2.6%) impressions; Location & Geography (4.4%); and Professional & Education (4.3%) relations. DialogRE exhibits several properties that make it particularly challenging and suitable for studying explainability–performance trade-offs: 96.0% of relations require cross-sentence reasoning, 89.9% involve at least one speaker entity, and the conversational style is characterized by high pronoun usage and low information density, which also complicates relation extraction in multi-turn contexts. These characteristics make DialogRE an effective benchmark for evaluating whether interpretable models can maintain competitive performance on complex, real-world conversational data.[3]

Relation Extraction Model: To obtain entity pairs and their contextual windows for emotion classification, we employ a DeBERTa-v3-large model fine-tuned on the DialogRE dataset. DeBERTa-v3-large has demonstrated strong performance on dialogue understanding tasks due to its disentangle attention mechanism and enhanced mask decoder (He et al., 2021). We fine-tune the model to identify subject–object entity pairs and extract their corresponding 3-turn conversational context (preceding turn, target turn, and following turn). These extracted tuples serve as structured inputs to all downstream emotion attribution models, enabling a controlled comparison across modeling paradigms.

Training Data: For neural model training, we adopt an LLM-assisted silver-labeling strategy. We employ Llama-3-8B-Instruct to annotate 1,179 instances from the DialogRE training split. The model is prompted with the full dialogue, the target entity pair, and explicit emotion-relation definitions.To guide the model's predictions, we include 150 manually annotated instances as in-context demonstrations.To ensure label quality, we retain only predictions with confidence scores ≥ 0.70. This process yields a balanced silver-labeled training set comprising 400 positive, 393 neutral, and 386 negative samples. We further reserve 20% of this data for validation during fine-tuning.

Test Set Construction: From the official DialogRE test split, we extracted 150 entity pairs using to ensure a controlled evaluation environment. To maintain focus on interpersonal dynamics, we filtered the data to include only recognized social relations, such as *per:friends*, *per:spouse*, and *per:girl/boyfriend*, while excluding non-social attributes like *per:age* or *per:title*. For each sampled pair, we extracted a 3-turn contextual dialogue window (comprising the target utterance and its immediate neighbors) to provide the situational evidence required for manual annotation and rule-based sentiment analysis. The final gold-standard distribution for this test set consists of 70 positive (46.7%), 44 negative (29.3%), and 36 neutral/unanswerable (24.0%) instances, providing a benchmark for comparing neural fine-tuning against our interpretable Aigents approach.

[3] `https://github.com/nlpdata/dialogre`

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

4 MODELS

We evaluate three distinct paradigms for emotion attribution, each representing a different point along the explainability-performance spectrum: rule-based pattern matching (Aigents), fine-tuned neural models (RoBERTa), and few-shot large language models (Llama-3).

4.1 RULE-BASED APPROACH: AIGENTS

Aigents is an interpretable n-gram matching system originally developed for sentiment analysis (Raheman et al., 2022). The model operates through dictionary-based pattern matching with predefined lexicons of emotional expressions. We adapt Aigents for dialogue-based emotion attribution by implementing cross-turn pattern aggregation with exponential strength normalization.

Architecture: Aigents employs a hierarchical n-gram matching strategy with the "priority on order" principle: longer n-grams take precedence over their constituent shorter n-grams. For instance, if the tetragram ["not", "a", "bad", "thing"] matches, all constituent n-grams including the bigram ["bad", "thing"] and unigram ["bad"] are disregarded. This prevents double-counting and allows the model to capture negation and multi-word expressions accurately. Similarly, matching the bigram ["no", "good"] disregards both constituent unigrams ["no"] and ["good"]. The system maintains separate lexicons for positive and negative emotional expressions. The positive lexicon comprises over 3,800 n-grams, and the negative lexicon contains over 8,200 n-grams,

Scoring Algorithm: For each dialogue turn, Aigents extracts all matching n-grams and computes separate positive (P) and negative (N) raw scores based on match frequencies. The model then applies an exponential strength transformation to normalize scores:

$$r = \max(P, |N|) \tag{1}$$

$$s = 1 - e^{-1.5 \cdot r} \tag{2}$$

where r represents the raw dominant score and s is the normalized strength in range [0, 1]. This exponential transformation maps raw scores as follows: 0.3 → 0.36, 0.5 → 0.53, 1.0 → 0.78, and 1.5 → 0.89. For texts containing multiple emotion words (total emotion > 1.0), we apply an optional multi-word boost:

$s' = \min(1.0, s + \min(0.15, (P + |N| - 1.0) \times 0.1))$.

Classification Decision: The model computes a margin score $m = |P - |N||$ and confidence $c = \frac{m}{P+|N|}$ when the denominator is non-zero. The final three-way classification uses threshold-based rules:

- **neutral** if $m < \tau_m$ and $r < \tau_r$
- **positive** if $P > |N|$ (among non-neutral cases)
- **negative** if $|N| > P$ (among non-neutral cases)

where $\tau_m = 0.2$ (margin threshold) and $\tau_r = 0.3$ (raw strength threshold) are empirically tuned on validation data. This confidence-based approach enables the inherently binary Aigents model to detect neutral cases when both positive and negative signals are weak or balanced.

Neutral Label Handling for Aigents Evaluation: Aigents performs binary classification (positive, negative), while gold labels include a neutral class. To enable fair comparison, we use three evaluation modes. *Mode 1* excludes gold neutral instances, evaluating binary performance on polarized samples only (N = 114). *Mode 2* maps neutral labels to the dorminant class class, enabling full dataset comparison (N = 150). *Mode 3* excludes samples identified as likely neutral based on weak lexical evidence, focusing on high-confidence predictions (N = 97). These complementary modes ensure fair evaluation while reflecting the architectural constraints of rule-based emotion attribution.

Contextual Integration: To capture the emotional nuances of multi-turn conversations, we analyze a 3-turn context window (preceding, target, and following utterances) as a unified situational block. Rather than applying arbitrary linear weights to each turn, we aggregate all n-gram matches across

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

the window into a total raw score R. This holistic approach ensures that supporting evidence (such as a prompt in a preceding turn or a reaction in a following turn) is captured with equal fidelity.

The aggregate signal is then transformed via an exponential saturation function:

$$S(R) = 1 - e^{-1.5R} \tag{3}$$

to determine the final relationship strength $S(R)$. This non-linear transformation prevents "evidence inflation" while maintaining the interpretability of the extracted emotional signals.

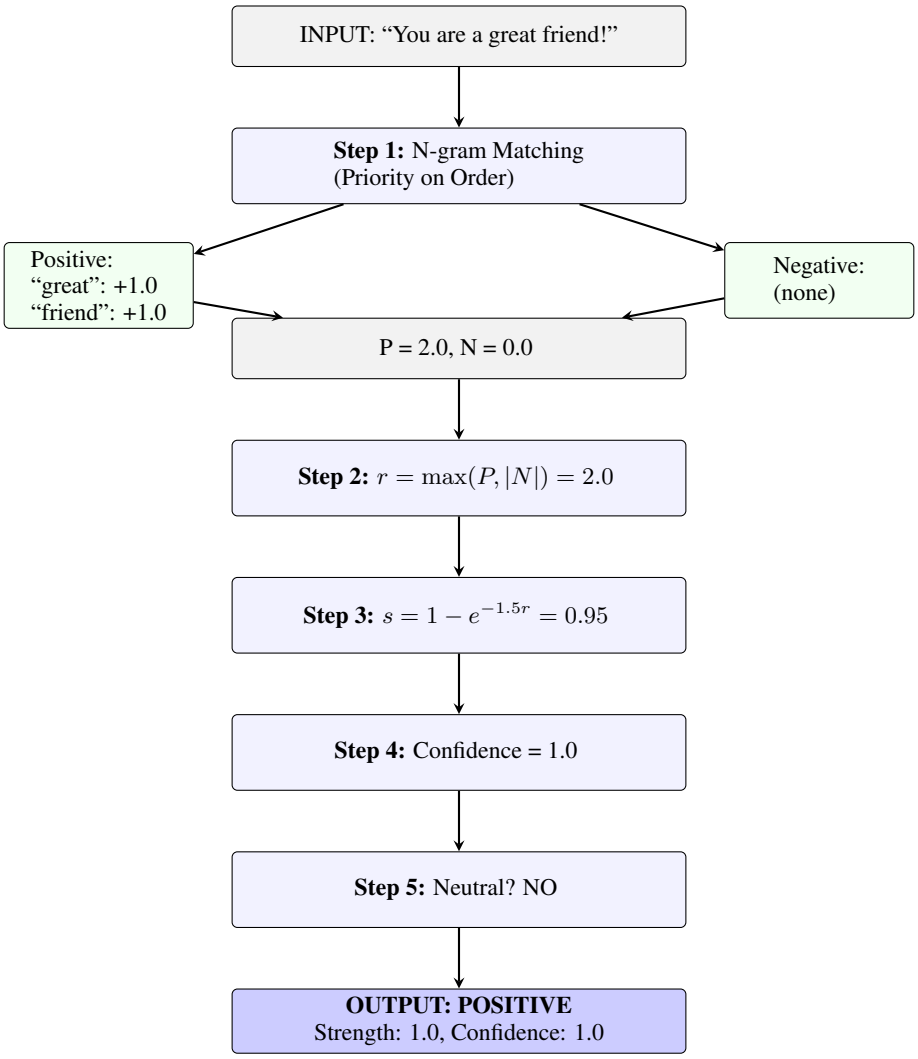


Figure 1: Interpretable emotion attribution pipeline of Aigents. The model applies ordered n-gram matching over positive and negative lexicons, followed by exponential strength normalization and confidence-based neutral detection, yielding a fully transparent polarity decision.

4.2 Fine-Tuned Neural Approach: RoBERTa:

We employ RoBERTa (Liu et al., 2019), a robustly optimized Transformer encoder pre-trained on 160GB of text with dynamic masking and extended training. To incorporate emotion-specific prior knowledge, We initialize from a RoBERTa-large checkpoint fine-tuned on GoEmotions (Demszky et al., 2020), publicly available via the HuggingFace Model Hub.[4], a large-scale emotion annotation

[4] https://huggingface.co/Laksssshya/RoBERTa-large-goemotions

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

dataset comprising approximately 58k Reddit comments labeled across 28 fine-grained emotions and a neutral class. The GoEmotions pretrained checkpoint we adopt was optimized for multi-label emotion classification using focal loss and label-specific threshold calibration to address class imbalance. While the original model supports multi-label prediction, we repurpose the encoder representations for single-label emotion attribution by replacing the original output layer with a task-specific classifier trained on DIALOGRE-derived emotion labels.

Context Variants: To assess the impact of conversational context, we evaluate two input configurations: ROBERTA-1TURN (`[CLS] target_turn [SEP]`) processes only the target utterance containing the subject–object relation, while ROBERTA-3TURN (`[CLS]` $\texttt{turn}_{-1}$ `[SEP]` $\texttt{turn}_{0}$ `[SEP]` $\texttt{turn}_{+1}$ `[SEP]`) incorporates a local conversational window consisting of the preceding, target, and following turns. Both variants use a maximum sequence length of 257 tokens, covering approximately 95% of DIALOGRE instances without truncation.

4.3 FEW-SHOT LLM APPROACH: LLAMA-3

We evaluate Llama-3-8B-Instruct (AI@Meta, 2024), an instruction-tuned large language model with 8 billion parameters trained on over 15 trillion tokens. We evaluate it using in-context few-shot learning with a baseline prompt consisting of detailed task instructions and 10 labeled examples. To determine whether performance limitations arise from sub-optimal prompting rather than inherent model constraints, we conduct a systematic ablation across three prompt templates and three shot counts (3, 5, and 10 examples), yielding 9 configurations. The evaluated prompt designs include: (1) a standard instruction-based template with explicit classification guidelines, (2) a chain-of-thought template encouraging step-by-step reasoning, and (3) a structured JSON-format template enforcing constrained outputs (Table 3).

5 EXPERIMENTAL SETUP

Relation Extraction (DeBERTa-v3-Large): For relation extraction on DialogRE, we fine-tune microsoft/deberta-v3-large model [5] with a maximum sequence length of 512. Due to memory constraints, we use a per-device batch size of 2 with gradient accumulation over 4 steps (effective batch size of 8). The model is trained for 5 epochs using a learning rate of 1×10^{-5}. To address class imbalance, we downsample 70% of *no_relation* instances and apply a reduced loss weight (0.5) to the *unanswerable* class. All experiments are conducted with fixed random seeds (42) to ensure reproducibility.

Neural Emotion Attribution (RoBERTa): As discussed in section 4, we fine-tune two variants of RoBERTa-large, initialized from the GoEmotions-specialized checkpoint.We employ the AdamW optimizer with a learning rate of 8×10^{-6} and 9×10^{-6} for 1turn and 3turns model respectively, weight decay of 0.01, and a batch size of 16. To prevent overfitting on the dialogue context, we implement early stopping with a patience of 3 epochs based on the validation Macro F1 score. Training is conducted on an NVIDIA T4 GPU using mixed-precision (FP16), typically reaching convergence within 3–5 epochs.

LLM-Based Data Augmentation (Llama-3): To generate silver labels, we employ Llama-3-8B-Instruct[6] with 4-bit NormalFloat4 (NF4) quantization and double quantization. This reduces the memory footprint from $\sim$15GB to $\sim$5.3GB. Inference is performed via greedy decoding (temperature = 0.1, max 256 tokens) with a 10-sample few-shot strategy.

Evaluation Metrics: We evaluate performance using the Macro-averaged F1 score to balance representation across positive, negative, and neutral classes. We report results for RoBERTa (1-turn), RoBERTa (3-turn), and Aigents. Statistical significance is verified via 95% confidence intervals derived from bootstrap resampling ($N = 1000$).

[5] `https://huggingface.co/microsoft/deberta-v3-large`
[6] `https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct`

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Table 1: Hyperparameter specifications across model architectures.

Hyperparameter	**RoBERTa (1T)**	**RoBERTa (3T)**	**Llama-3**	**Aigents**
Model:				
Base	RoBERTa-large	RoBERTa-large	Llama-3-8B-Instruct	N-gram Lexicon
Init	GoEmotions FT	GoEmotions FT	Meta	Pre-built
Quant.	None	None	4-bit NF4	None
Max Len	128	256	2048	N/A
Training:				
Optimizer	AdamW	AdamW	N/A	Deterministic
LR	8×10^{-6}	9×10^{-6}	N/A	N/A
WD	0.01	0.01	N/A	N/A
Batch	16	16	1	N/A
Epochs	15	15	N/A	N/A
Logic:				
Temp	N/A	N/A	0.1	N/A
Formula	Linear	Linear	N/A	$1 - e^{-1.5R}$
Thresh	Softmax	Softmax	N/A	0.4
Env:				
Hardware	T4 (Colab)	T4 (Colab)	T4 (Colab)	CPU (Docker)
Precision	FP16	FP16	4-bit	FP32

Table 2: Overall performance comparison of rule-based (Aigents), fine-tuned transformer (RoBERTa), and few-shot LLM (Llama-3) approaches for emotion attribution in social graphs. RoBERTa-3turn achieves the highest Macro F1 (0.636), demonstrating the benefit of multi-turn conversational context. The interpretable Aigents system achieves comparable performance (Macro F1 up to 0.58, 0.6236, 0.625 when neutral are mapped to dorminant class, when gold neutral are excluded, and when detected neutral are excluded respectively). Few-shot Llama-3 exhibits substantially lower and less stable performance (Macro F1 0.528). Confidence intervals (95%) are computed via bootstrap resampling ($N = 1000$).

Model	**N**	**Acc**	**F1 Pos**	**F1 Neg**	**F1 Neu**	**F1 Macro [CI]**
RoBERTa-3turn	150	0.653	0.696	**0.719**	0.493	**0.636 [0.554, 0.715]**
Aigents (Mode 3)††	97	**0.701**	**0.794**	0.453	—	0.624 [0.516, 0.727]
Aigents (Mode 2)†	150	0.593	0.651	0.512	*mapped*	0.582 [0.500, 0.659]
Aigents (Mode 1)	114	0.667	0.750	0.500	—	0.625 [0.526, 0.712]
Llama-3§	150	0.567	0.642	0.537	0.407	0.528 [0.484, 0.568]
RoBERTa-1turn	150	0.633	0.676	0.638	**0.537**	0.617[0.579, 0.655]
RoBERTa-baseline	150	0.573	0.708	0.500	0.405	0.538 [0.496, 0.580]

6 Results

6.1 Overall Performance

Llama-3-8B achieves F1 macro = 0.528, substantially lower than RoBERTa-3turn (0.636), indicating that few-shot prompting without task-specific fine-tuning provides insufficient performance gain. Ablation across 9 configurations (Table 3, Appendix) shows performance ranges from F1 = 0.501 (best: Standard template, 3-shot) to F1 = 0.294 (Chain-of-Thought, 5-shot), indicating that prompt engineering provides marginal improvements but cannot overcome inherent LLM limitations on this task.

6.2 Statistical Significance

We assess statistical reliability using 95% bootstrap confidence intervals computed from 1000 resamples (Table 2). RoBERTa-3turn achieves the highest macro F1 (0.636, CI: [0.554, 0.715]), slightly outperforming RoBERTa-1turn (CI: [0.579, 0.655]). The small overlap between inter-

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

vals suggests a modest but consistent benefit from incorporating multi-turn conversational context. In contrast, RoBERTa-3turn substantially outperforms the few-shot Llama-3 model (CI: [0.484, 0.568]), with non-overlapping confidence intervals indicating a statistically significant difference ($p < 0.05$). Further analysis of nine prompting configurations for Llama-3 reveals high sensitivity to prompt design (F1 range: 0.294–0.501), yet even the best configuration remains well below RoBERTa-3turn and the interpretable Aigents Mode 3 system (F1 = 0.624). This gap, combined with performance degradation under alternative prompt strategies (e.g., Chain-of-Thought mean F1 = 0.328), highlights inherent limitations of few-shot prompting for structured emotion classification compared to task-specific fine-tuning and interpretable rule-based methods. Finally, all advanced approaches substantially outperform the baseline RoBERTa model, confirming the importance of multi-turn context and task-specific adaptation.These results challenge the conventional assumption that explainability necessarily entails performance trade-offs: Aigents achieves competitive performance with RoBERTa while providing complete transparency.

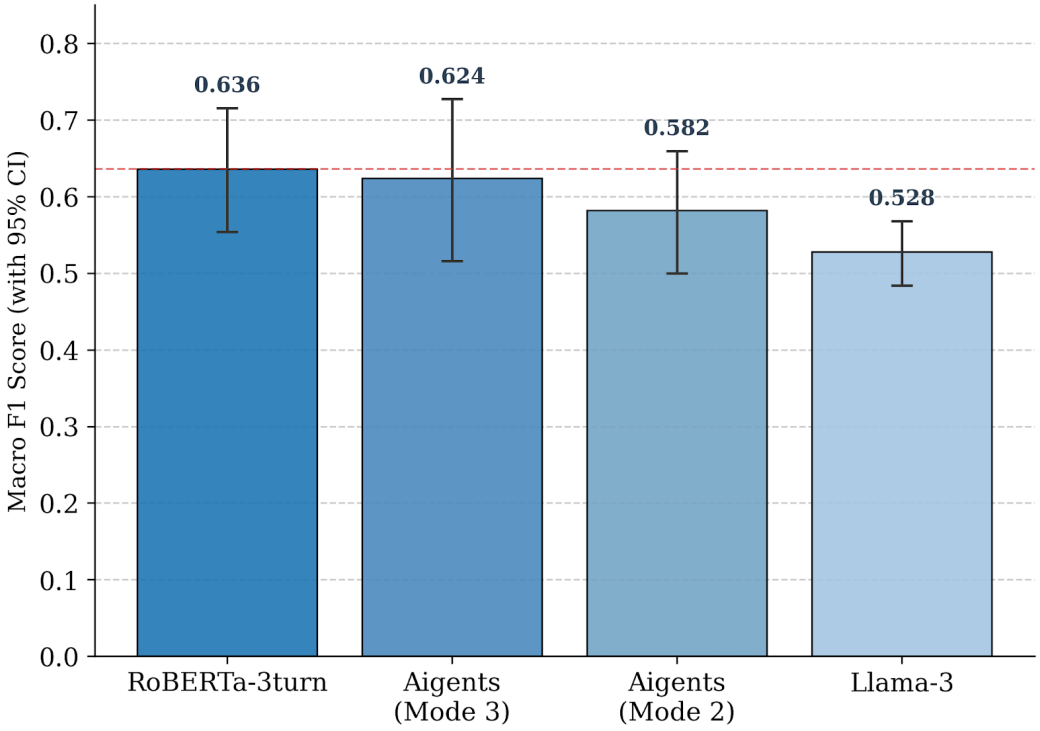


Figure 2: Overall model performance Macro F1 with 95% Bootstrap Confidence Intervals [CI]. Bold indicates the highest value in each metric. Aigents (mode 2) 3-class mode where Neutral instances were mapped to dorminant, Aigents (Mode 3) binary mode where detected neutrals excluded, and Llama-3 8B few-shot performance. The RoBERTa-3turn model achieves the highest Macro F1, while Aigents (Mode 3) demonstrates superior precision on its targeted subset.

.

7 Discussion

Our results show that interpretability can be achieved without sacrificing performance. The interpretable Aigents system attains F1 = 0.582 (mapped) and F1 = 0.624 (detected neutral excluded) on binary emotion classification, statistically comparable to RoBERTa-3turn (F1 = 0.636) based on overlapping confidence intervals. Unlike neural models, Aigents provides explicit, human-readable

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

justifications through matched lexical patterns and deterministic weights, making its predictions directly auditable. However, Aigents does not support neutral emotion attribution, which constitutes 24% of the dataset, highlighting a key limitation of rule-based approaches in modeling absence-based categories. Consequently, interpretable systems are well-suited for transparent binary attribution, while fine-tuned neural models remain necessary for full multi-class coverage. Incorporating multi-turn conversational context yields a modest improvement over single-turn RoBERTa, with confidence intervals indicating a small but consistent benefit. This suggests that while most emotion cues reside within the target utterance, local conversational context provides additional disambiguating information in some cases. Given the increased computational cost of multi-turn inputs, this improvement represents a favorable trade-off in accuracy-critical applications.

Few-shot prompting with Llama-3 substantially underperforms both fine-tuned and rule-based approaches. Despite systematic evaluation across nine prompt configurations, Llama-3 achieves a maximum F1 of 0.528 (10 few-shot) and 0.501 with ablation, with confidence intervals significantly below RoBERTa-3turn and consistently lower than Aigents. The large performance variance across prompts (F1 range: 0.294–0.501) indicates sensitivity to prompt design, but even optimal prompting fails to close the gap. Notably, the interpretable Aigents system outperforms Llama-3 by approximately 0.12 F1, demonstrating that transparent, rule-based methods can exceed few-shot large language models on structured emotion attribution tasks where explicit lexical cues dominate. This suggests that the limitation lies in the few-shot paradigm itself rather than prompt engineering, and that task-specific fine-tuning is necessary for neural models to achieve competitive performance. Overall, these findings highlight three key conclusions: (1) interpretable rule-based systems can achieve performance comparable to fine-tuned neural models in binary settings, (2) multi-turn conversational context provides modest but consistent improvements, and (3) few-shot prompting alone is insufficient for high-accuracy emotion attribution without task-specific adaptation.

8 Limitations

Several limitations should be acknowledged. First, the evaluation is conducted on a relatively small test set (N = 150), which, while sufficient for bootstrap-based statistical comparison, limits the strength of generalization claims. Furthermore, the data are derived from scripted *Friends* dialogues, which may over-represent explicit emotional expressions and structured conversational patterns compared to spontaneous real-world interactions. Moreover, the Aigents system does not natively model neutral emotion, as its rule-based architecture detects emotions through the presence of explicit lexical cues. To enable fair three-class comparison, neutral instances were mapped using confidence-based procedures (Mode 2), which introduces an approximation and may not fully capture neutrality as an independent semantic category. Additionally, although silver labels were filtered using confidence thresholds and partial manual validation, residual labeling bias from the LLM-assisted annotation process may persist and influence model evaluation. Finally, few-shot Llama-3 evaluation, while systematically examined across nine prompt configurations, remains constrained by the inherent variability and instability of prompt-based inference. Although our ablation improves robustness of conclusions, it does not represent an upper bound achievable through full task-specific fine-tuning of large language models.

9 Conclusion

This work provides a controlled comparison of interpretable rule-based, fine-tuned neural, and few-shot LLM approaches for dialogue emotion attribution. On the shared 150-sample test set, RoBERTa-3turn achieves the highest overall performance (F1 = 0.636), while the interpretable Aigents system remains competitive (F1 = 0.582 in 3-class mode; F1 = 0.624 on high-confidence predictions), demonstrating that transparent rule-based models can approach neural performance when lexical emotion cues are explicit. In contrast, systematic ablation across nine prompting configurations shows that few-shot Llama-3 reaches a maximum F1 of 0.501 and 0.528 (10 few-shot only), substantially below both Aigents and RoBERTa, indicating inherent limitations of few-shot prompting for structured emotion classification. These findings highlight a practical trade-off: rule-based systems offer full interpretability and competitive binary performance, while fine-tuned neural models provide superior multi-class coverage. Future work will explore hybrid architectures com-

Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

bining interpretable lexical reasoning with neural handling of neutral and ambiguous emotional expressions.

REFERENCES

AI@Meta. Introducing llama 3: The most capable openly available llm to date, 2024. URL https://ai.meta.com/blog/meta-llama-3/. Accessed: 2024-04-18.

Niels Dekker, Tobias Kuhn, and Marieke van Erp. Evaluating named entity recognition tools for extracting social networks from novels. *PeerJ Computer Science*, 01 2019. ISSN 23765992. doi: 10.7717/peerj-cs.189. URL https://peerj.com/articles/cs-189/.

Dorottya Demszky, Dana Movshovitz-Attias, and other. Goemotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4040–4054, 2020. URL https://aclanthology.org/2020.acl-main.372/.

Xiachong Feng, Xiaocheng Feng, et al. Language model as an annotator: Exploring dialogpt for dialogue summarization. *aclanthology*, 2021. doi: 10.18653/v1/2021.acl-long.117. URL https://aclanthology.org/2021.acl-long.117/.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint*, 2021. URL https://arxiv.org/abs/2111.09543.

Jean Kaddour and Qi Liu. Synthetic data generation in low-resource settings via fine-tuning of large language models. *arXiv (Cornell University)*, 10 2023. doi: 10.48550/arxiv.2310.01119. URL http://arxiv.org/abs/2310.01119.

Maryam Khalid and Akane Sano. Exploiting social graph networks for emotion prediction. *Scientific Reports*, 13, 04 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-32825-9. URL https://doi.org/10.1038/s41598-023-32825-9.

Bharti Khemani, Shruti Patil, Ketan Kotecha, and Sudeep Tanwar. A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, 2024. ISSN 21961115. doi: 10.1186/s40537-023-00876-4. URL https://link.springer.com/article/10.1186/s40537-023-00876-4.

Bharti Khemani, Shruti Patil, Sachin Malave, and Jaya Gupta. Improved graph convolutional network for emotion analysis in social media text. *MethodsX*, 14:103325–103325, 04 2025. ISSN 2215-0161. doi: 10.1016/j.mex.2025.103325. URL https://doi.org/10.1016/j.mex.2025.103325.

Taewoon Kim and Piek Vossen. Emoberta: Speaker-aware emotion recognition in conversation with roberta. *arXiv preprint arXiv:2108.12009*, 2021. doi: 10.48550/arXiv.2108.12009. URL https://doi.org/10.48550/arXiv.2108.12009.

Chaitanya Kumar Kolli. Hybrid neuro-symbolic models for ethical ai in risk-sensitive domains. *arXiv (Cornell University)*, 11 2025. doi: 10.48550/arxiv.2511.17644. URL https://doi.org/10.48550/arxiv.2511.17644.

Anton Kolonin and Anna Arinicheva. Interpretable recognition of cognitive distortions in natural language texts. *arXiv preprint arXiv:2511.05969*, 2025. URL https://arxiv.org/abs/2511.05969.

Christopher Lalk, Kim Targan, et al. Employing large language models for emotion detection in psychotherapy transcripts. *Frontiers in Psychiatry*, 2025. ISSN 16640640. doi: 10.3389/fpsyt.2025.1504306. URL https://www.frontiersin.org/journals/psychiatry/articles/10.3389/fpsyt.2025.1504306/full.

J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977. doi: 10.2307/2529310. URL https://www.jstor.org/stable/2529310.

Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Xiao Liu, Jian Zhang, Heng Zhang, et al. Hierarchical dialogue understanding with special tokens and turn-level attention. *arXiv preprint arXiv:2305.00262*, 2023. doi: 10.48550/arXiv.2305.00262. URL https://doi.org/10.48550/arXiv.2305.00262.

Yinhan Liu, Myle Ott, et al. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/abs/1907.11692.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017. URL https://arxiv.org/abs/1705.07874.

Hadi Mohammadi, Ayoub Bagheri, Anastasia Giachanou, and Daniel L. Oberski. Explainability in practice: A survey of explainable nlp across various domains. *arXiv preprint arXiv:2502.00837*, 02 2025. doi: 10.48550/arxiv.2502.00837. URL http://arxiv.org/abs/2502.00837.

Ali Raheman et al. Aigents: Multi-agent systems for social graph analysis and emotion prediction. *arXiv preprint arXiv:2204.10185*, 2022. URL https://arxiv.org/abs/2204.10185.

Daking Rai, Yilun Zhou, Shi Feng, Abulhair Saparov, and Ziyu Yao. A practical review of mechanistic interpretability for transformer-based language models. *arXiv (Cornell University)*, 01 2024. doi: 10.48550/ARXIV.2407.02646. URL https://arxiv.org/abs/2407.02646.

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 97–101, 2016. doi: 10.18653/v1/n16-3020. URL https://doi.org/10.18653/v1/n16-3020.

Ahmed Salih, Zahra Raisi-Estabragh, and other. A perspective on explainable artificial intelligence methods: Shap and lime. *Advanced Intelligent Systems*, 7, 06 2024. ISSN 2640-4567. doi: 10.1002/aisy.202400304. URL https://doi.org/10.1002/aisy.202400304.

Writuraj Sarma. Explainability in nlp: Interpretable ai models for text classification and sentiment analysis. *International Journal of Communication Networks and Information Security (IJCNIS)*, 11(3):448–464, 2019. URL https://www.ijcnis.org/index.php/ijcnis/article/view/8056.

Lee Sharkey, Bilal Chughtai, et al. Open problems in mechanistic interpretability. *arXiv preprint arXiv:2501.16496*, 2025. URL https://arxiv.org/abs/2501.16496.

Weizhou Shen et al. Dialogxl: All-in-one xlnet for multi-party conversation emotion recognition. *arXiv preprint arXiv:2105.12907*, 2021. URL https://arxiv.org/abs/2105.12907.

Ala N. Tak et al. Interpretability in large language models. *arXiv preprint arXiv:2502.05489*, 2025. URL https://arxiv.org/abs/2502.05489.

Hailu Tan, Yan Liu, Xinying Liu, Lianyu Hu, and Zengyou He. Interpretable link prediction. *Chaos, Solitons and Fractals*, 2025. ISSN 09600779. doi: 10.1016/j.chaos.2024.115928. URL https://www.sciencedirect.com/science/article/abs/pii/S0960077924014802.

Dian Yu, Kai Sun, Claire Cardie, and Dong Yu. Dialogue-based relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4927–4940. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.444. URL https://aclanthology.org/2020.acl-main.444.

Ruoyu Zhang, Yanzeng Li, et al. Llmaaa: Making large language models as active annotators. *emnlp*, 2023a. doi: 10.18653/v1/2023.findings-emnlp.872. URL https://doi.org/10.18653/v1/2023.findings-emnlp.872.

Yazhou Zhang, Mengyao Wang, et al. Dialoguellm: Context and emotion knowledge-tuned large language models for emotion recognition in conversations. *arXiv (Cornell University)*, 01 2023b. URL https://arxiv.org/abs/2310.11374.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

A Appendix

B Training Dynamics Analysis

The RoBERTa-3turn model exhibits standard fine-tuning behavior: rapid optimization during early training (steps 0–150), followed by convergence between steps 150–300. Validation loss attains its minimum at step 300, after which performance degrades, and early stopping is therefore applied to prevent overfitting. At this point, the model achieves a macro-F1 of 0.809, with particularly strong performance on positive emotion (F1 = 0.845). Neutral emotion remains the most challenging class (F1 = 0.779), suggesting that additional conversational context can introduce competing emotional cues rather than resolve absence-based sentiment. The substantial macro-F1 improvement from initialization (+0.435) highlights effective transfer from GoEmotions pre-training, while post-convergence instability indicates sensitivity to noise in multi-turn dialogue context.



Figure 3: Training dynamics of RoBERTa-3turn for dialogue emotion classification. Training loss decreases rapidly and stabilizes by step 300. Validation loss reaches its minimum at step 300 (marked), after which overfitting emerges, motivating early stopping. Macro-F1 improves steadily (+0.435 overall), with strongest gains for positive emotion. Per-class trajectories indicate stable learning for positive and negative classes, while neutral performance remains comparatively volatile, reflecting contextual ambiguity introduced by multi-turn inputs.

C Results Details

C.1 Confusion Matrix Analysis

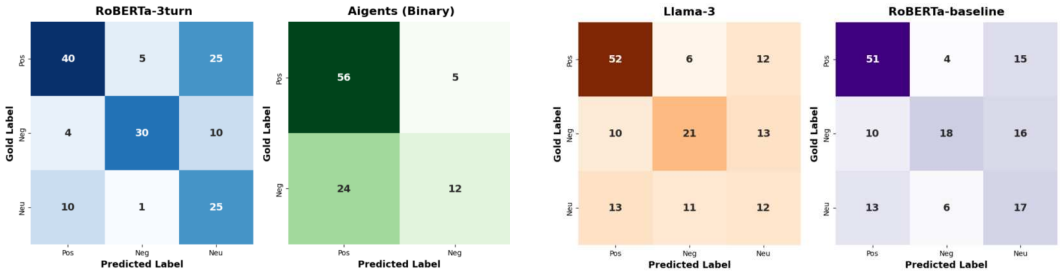


Figure 4: Comparison of model performance and human annotation consistency. (Left) Confusion matrix for the RoBERTa-3turn model. (Right) Inter-annotator agreement matrix ($\kappa = 0.579$) illustrating the distribution of concordant and discordant labels across the 150-sample gold standard.

Figure 4 summarizes confusion matrices for all evaluated models. Across paradigms, the dominant error pattern involves confusion between *neutral* and *positive* classes. In particular, neutral

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

utterances are frequently predicted as positive (e.g., 13/36 neutral instances for the RoBERTa baseline), reflecting the difficulty of distinguishing polite acknowledgment or factual engagement from genuine positive affect (e.g., "That's interesting"). Confusion between *negative* and *neutral* is less frequent (6–10 cases per model), consistent with negative emotions being more lexically explicit. Direct *positive–negative* confusion is rare (4–10 cases), indicating that emotional polarity is generally well captured when affect is overt; most errors instead involve neutral as an intermediate category.

On binary evaluation, Aigents exhibits asymmetric error behavior, with more false positives (negative→positive) than false negatives. This reflects lexicon sensitivity rather than contextual reasoning, leading to over-triggering in ambiguous cases and missed detections when sentiment is implicit.

C.2 Per-Class Analysis

Figure 2 summarizes per-class F1 scores across models, revealing consistent trends. **Positive emotion** is the easiest class for all approaches (F1: 0.642–0.794), reflecting the prevalence of explicit positive lexical cues in dialogue (e.g., expressions of appreciation or affiliation). Aigents (Mode 3) achieves the highest positive F1 (0.794) on the binary subset, indicating that rule-based lexicons effectively capture common positive patterns. **Negative emotion** shows moderate performance (F1: 0.500–0.719), with RoBERTa-3turn outperforming other models. This suggests that multi-turn context aids the disambiguation of criticism and dissatisfaction from neutral statements, which are often lexically similar but pragmatically distinct. **Neutral emotion** remains the most challenging class for all three-class models (F1: 0.405–0.537), with RoBERTa-baseline performing worst (0.414). This difficulty reflects the absence-based nature of neutrality, which lacks distinctive surface markers and is subject to annotation ambiguity. Precision–recall patterns further differentiate models. Aigents (Mode 3) exhibits high positive recall but lower precision, consistent with aggressive lexicon matching. RoBERTa-3turn maintains relatively balanced precision and recall across classes, while Llama-3 shows low neutral recall, indicating a tendency to default to polarized labels under uncertainty.

C.3 Error Analysis

We manually inspected 30 misclassified instances (10 per model) to identify recurrent failure modes.

Implicit Emotion (42%): All models struggle when affect is conveyed pragmatically rather than lexically. For example, "Why isn't Janine coming?" is labeled neutral despite an implied accusatory tone, and "Wow, really?" is ambiguous between enthusiasm and skepticism.

Sarcasm and Irony (23%): Utterances where surface polarity contradicts intended meaning (e.g., "Oh that's just great.") consistently cause errors across models, highlighting the limitations of both pattern-based and contextual encoders.

Neutral–Positive Boundary (35%): Polite or transactional expressions (e.g., "Thanks for the information") are frequently misclassified as positive, underscoring annotation ambiguity and the absence-based nature of neutrality.

Model-Specific Trends: Aigents fails when cross-turn or third-party context is required for correct attribution. RoBERTa occasionally overgeneralizes from lexical associations learned during pre-training (e.g., "Oh God"), while Llama-3 tends to assign polarized labels to ambiguous cases, likely reflecting bias toward emotionally salient content in its pre-training data.

D Aigents Graph Construction

Figure 5 visualizes the dialogue-level social graph constructed from Aigents for a 10-turn conversation. Nodes represent dialogue participants, while directed edges encode inferred emotional attribution between speakers across turns. Edge color indicates polarity (green for positive, red for negative), and edge labels denote the corresponding confidence or strength score produced by the rule-based pipeline. This graph formulation makes explicit how emotional interactions accumulate across dialogue turns, allowing inspection of asymmetric or conflicting attributions between speaker

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

pairs. Unlike neural encoders such as RoBERTa, which implicitly model interaction structure within latent representations, Aigents exposes the interaction graph directly, enabling transparent analysis of emotion flow and speaker-to-speaker dynamics. In contrast, our RoBERTa models yield a structurally similar interaction pattern but additionally assign a neutral label, which is omitted here due to Aigents' binary formulation.

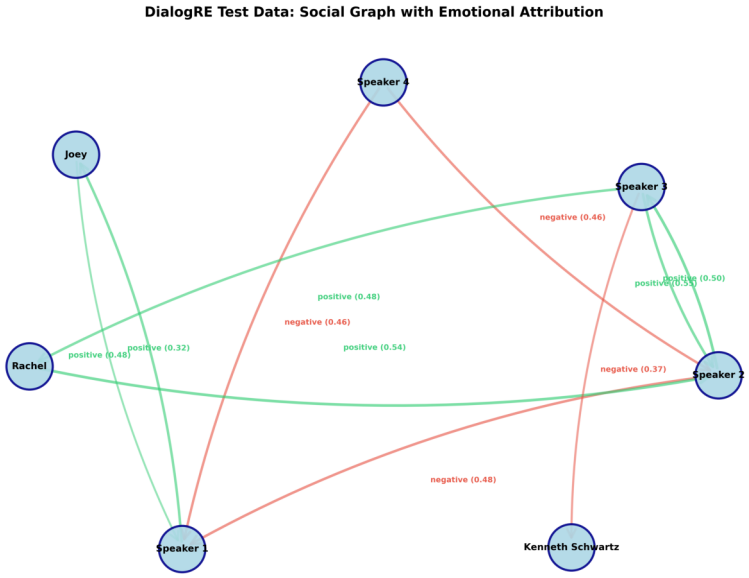


Figure 5: Dialogue-level social graph induced by Aigents for a 10-turn conversation. Nodes denote speakers and edges represent directed emotion attributions, and color indicating polarity (positive/negative) and edge labels showing attribution scores and polarity.

E Inter-Annotator Agreement

To validate our evaluation benchmark, two independent annotators (with English proficiency and background in affective computing) labeled a 150-sample subset of the DialogRE dataset. We achieved a Cohen's Kappa of $\kappa = 0.579$ (see 6), representing "moderate" agreement according to standard interpretability scales Landis & Koch (1977). The disagreements, primarily centered on the distinction between neutral and subtle negative affect, indicating the inherent subjectivity of social-emotional relation extraction. All discordant instances were subsequently resolved through a third-party adjudication process to establish a final consensus gold standard for model benchmarking.

F Ablation Results

The Llama-3 ablation study (Table 3) reveals fundamental constraints on few-shot performance. The performance range across 9 configurations (F1 = 0.294 to 0.501, span = 0.207) demonstrates significant sensitivity to prompting strategy, yet even optimal configuration achieves only F1 = 0.501—substantially below both RoBERTa-3turn (F1 = 0.636) and Aigents Mode 3 (F1 = 0.624). This performance gap, combined with the degradation observed in alternative prompt templates (Chain-of-Thought mean = 0.328), suggests that few-shot prompting fundamentally underperforms task-specific fine-tuning or interpretable rule-based approaches on structured emotion classification tasks.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1013 DOI: 10.66693/mathai.1013
Special Issue: MathAI 2026 Selected Papers
Received: 11 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

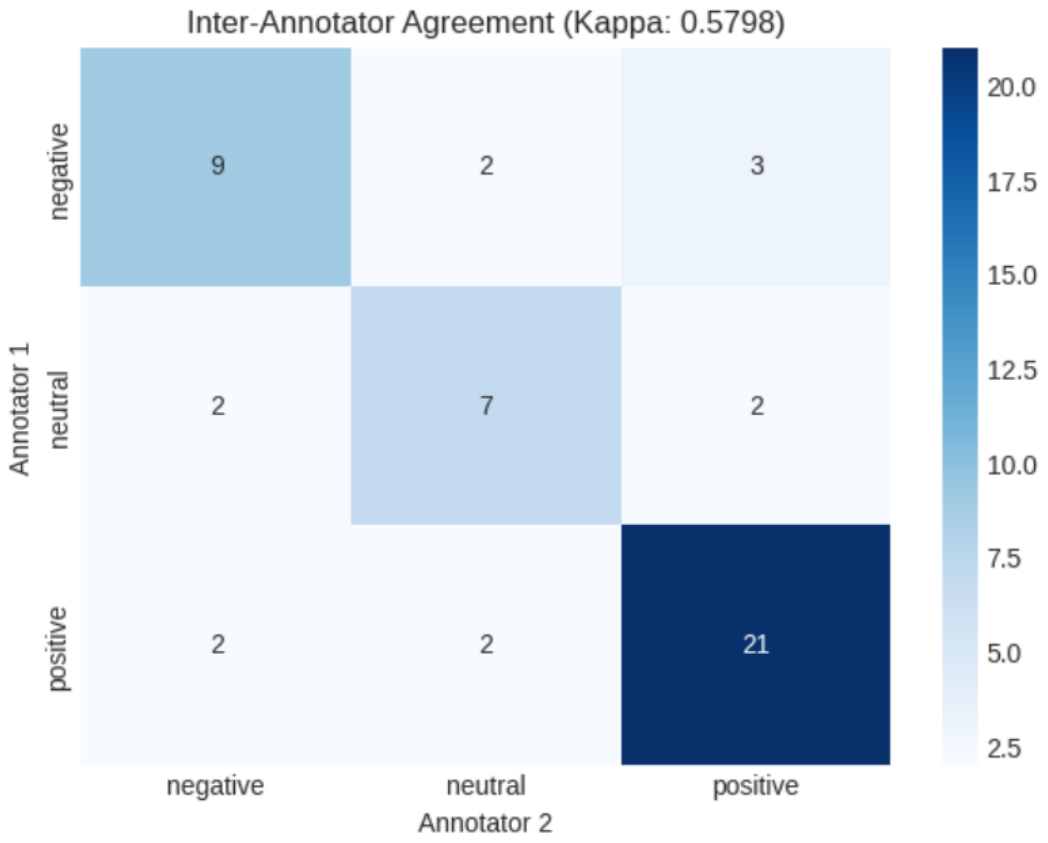


Figure 6: Inter-annotator agreement results with Cohen's Kappa (κ) interpretation. Our achieved $\kappa = 0.57$ falls in the "moderate" agreement range (0.41–0.60), validating annotation reliability.

Table 3: Llama-3-8B prompt engineering ablation study: F1 macro scores across 3 prompt templates and 3 shot counts (9 configurations total). Baseline configuration (Template 1, 10-shot) achieves F1 = 0.488. Best configuration (Template 1, 3-shot) achieves F1 = 0.501. Results demonstrate limited improvement potential through prompt engineering alone. All configurations evaluated on identical 150-sample test set.

Template	**3-shot**	**5-shot**	**10-shot**	**Mean**
1. Standard	0.501	0.447	0.488	0.479
2. Chain-of-Thought	0.394	0.296	0.294	0.328
3. JSON-Format	0.382	0.356	0.390	0.376
Mean (all)	0.426	0.366	0.391	

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

HYBRID BI-LEVEL INDEX FOR SHORTEST PATHS IN TEMPORAL NETWORKS

Vasilev M.
Moscow State University
s02230030@gse.cs.msu.ru

ABSTRACT

Temporal graphs provide a natural model for dynamic relational data arising in modern AI systems, including event streams, temporal knowledge graphs, interaction networks, and transaction systems. Efficient reachability querying in such graphs constitutes a fundamental operation underlying temporal reasoning, feature extraction, and dynamic graph learning.

In this paper, we propose a parameterized hybrid indexing framework for temporal reachability queries. Vertices are adaptively partitioned into two classes depending on the size of their reachable sets, enabling a controllable trade-off between memory usage and query time. Assuming a power-law degree distribution, we derive an analytical model for the proportion of promoted (large) vertices as a function of a promotion threshold. Closed-form asymptotic estimates for memory consumption and expected query time are obtained. We further prove the existence of a unique optimal threshold minimizing a combined memory–time cost functional.

Theoretical predictions are validated experimentally, revealing a characteristic U-shaped dependence of query time on the promotion parameter. The results provide a mathematically grounded foundation for adaptive indexing in large-scale temporal graph analytics and AI-driven dynamic data systems.

1 INTRODUCTION

Temporal graphs model systems in which interactions occur at specific time instances. Such structures naturally arise in communication networks, social interactions, financial transaction streams, logistics systems, and event-based knowledge graphs Wu et al. (2014); Casteigts et al. (2024). In contrast to static graphs, the validity of a path depends not only on connectivity but also on temporal consistency of edge timestamps. This additional constraint fundamentally alters algorithmic complexity and limits the direct applicability of classical graph algorithms Casteigts et al. (2021); Bumpus & Meeks (2022).

Beyond traditional network analysis, temporal graphs have become increasingly important in AI applications. Modern learning systems operate on dynamic relational data, including temporal knowledge graphs, streaming interaction networks, and evolving dependency structures. Operations such as temporal reachability, time-respecting path extraction, and dynamic influence propagation form essential building blocks for representation learning, temporal reasoning, and graph-based inference pipelines. Consequently, scalable temporal reachability indexing plays a critical role in AI infrastructures processing large-scale event data.

A substantial body of work addresses path and reachability problems in temporal graphs. Wu et al. Wu et al. (2014) formalize earliest-arrival and fastest-path problems, demonstrating that temporal constraints significantly complicate classical formulations. Casteigts et al. Casteigts et al. (2024) analyze different semantics of temporal reachability (strict, simple, proper, happy), highlighting the necessity of precise formal models. Further extensions incorporate waiting-time constraints and structural restrictions Casteigts et al. (2021); Bumpus & Meeks (2022). Large-scale and distributed solutions have also been proposed Zhang et al. (2019); Chen et al. (2019); Ding et al. (2021).

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

For static graphs, reachability indexing via labeling schemes has proven highly effective. Notable examples include 2-hop labeling methods Cohen et al. (2003), hub-based labeling techniques Abraham et al. (2010), and scalable label construction algorithms Ding et al. (2008). However, incorporating temporal constraints into such frameworks is nontrivial due to the combinatorial explosion of time-respecting paths.

An additional challenge arises from the structural properties of real-world networks. Empirical studies show that many large-scale graphs follow power-law degree distributions Barabási & Albert (1999); Clauset et al. (2009). The presence of highly connected hubs leads to strong heterogeneity in reachable-set sizes and temporal influence Kim & Anderson (2011). Ignoring this structural imbalance may result in inefficient memory usage or degraded query performance.

In this work, we introduce a hybrid indexing scheme that explicitly exploits structural heterogeneity. Vertices are partitioned according to a promotion threshold that depends on the size of their reachable sets. Small vertices store precomputed reachability information, while large vertices rely on bounded exploration at query time. The promotion parameter serves as a tunable control governing the trade-off between memory consumption and query latency.

Our contributions are as follows:

- We derive an analytical expression for the fraction of promoted vertices under a power-law degree model.
- We obtain asymptotic estimates for memory usage and expected query time as explicit functions of the promotion threshold.
- We prove the existence of a unique optimal threshold minimizing a combined memory–time cost functional.
- We empirically validate the theoretical predictions and explain the observed U-shaped behavior of query time through architectural effects.

The proposed framework provides a mathematically grounded perspective on adaptive indexing in temporal graphs. It bridges structural graph theory, asymptotic analysis, and scalable algorithm design, and is directly applicable to AI systems operating on dynamic relational data.

2 Temporal Graph Model

We consider a directed *temporal graph*

$$G = (V, E),$$

where each edge is represented by a time-stamped event

$$(u, v, t),$$

indicating that a transition from vertex u to vertex v is possible at time t Wu et al. (2014); Casteigts et al. (2024; 2021). We assume that events arrive in non-decreasing order of time, which reflects streaming dynamic systems in which the chronological order of interactions is essential Bumpus & Meeks (2022).

A temporal (time-respecting) path is a sequence of edges

$$(u_0, u_1, t_1), (u_1, u_2, t_2), \ldots, (u_{k-1}, u_k, t_k)$$

such that

$$t_1 \leq t_2 \leq \cdots \leq t_k.$$

The earliest-arrival distance from u to v is the minimum time t such that v is reachable from u by a temporal path arriving at time t Wu et al. (2014). This metric plays a central role in temporal reachability analysis and dynamic path planning.

Computing such paths efficiently is nontrivial. Wu et al. Wu et al. (2014) study earliest-arrival and fastest paths, demonstrating that straightforward adaptations of static graph traversal are often inefficient in high-volume event streams. Distributed solutions for large-scale temporal graphs have been

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

proposed in Zhang et al. (2019), while labeling and hub-based indexing techniques have proven effective in static settings Ding et al. (2008); Cohen et al. (2003); Abraham et al. (2010). However, extending these approaches to temporal graphs requires explicit handling of time constraints, significantly increasing computational and storage costs.

Real-world networks frequently exhibit heavy-tailed degree distributions Barabási & Albert (1999); Clauset et al. (2009). Such heterogeneity leads to large variability in reachable-set sizes and temporal influence Kim & Anderson (2011). Therefore, indexing strategies must account for structural imbalance in order to achieve scalable performance.

The adopted model captures the essential aspects of temporal dynamics and serves as a foundation for constructing adaptive hybrid indices for reachability queries in large, structurally heterogeneous networks.

3 Hybrid Temporal Reachability Index

3.1 Problem Formulation

Given a stream of temporal edges, we aim to maintain a data structure that supports efficient earliest-arrival distance queries between vertices. A temporal path is valid only if timestamps along the path are non-decreasing Wu et al. (2014); Casteigts et al. (2021).

Classical indexing approaches such as 2-hop labeling Cohen et al. (2003), hub-based labeling Abraham et al. (2010), and distributed reachability frameworks Zhang et al. (2019) achieve strong performance in static graphs. However, their direct application to dynamic temporal graphs is hindered by high storage overhead and the need to encode temporal constraints explicitly.

3.2 Main Idea

The key observation is that vertices differ significantly in the size of their reachable predecessor sets Kim & Anderson (2011); Bumpus & Meeks (2022). This structural heterogeneity motivates an adaptive strategy:

- For vertices with small reachable sets, it is efficient to store complete reachability tables.
- For vertices with large reachable sets, full storage becomes memory-intensive and unnecessary.

Accordingly, vertices are partitioned into two classes: **small** and **large**. The boundary between them is determined by a promotion threshold B.

3.3 Stored Data Structures

For each vertex v, we maintain:

- direct_edges(v) — the set of incoming temporal edges;
- dist_all(v) — a mapping to the vertices with the time of the earliest arrival, representing complete information about reachability Ding et al. (2008); Cohen et al. (2003).

For **small** vertices, both dist_all(v) and direct_edges(v) are stored. For **large** vertices, only direct_edges(v) are retained. Thus, large vertices keep only local structural information, reducing memory consumption.

3.4 Event Processing

Upon receiving a new event

$$(u \to v, t),$$

the algorithm performs the following steps:

1. Insert the edge into direct_edges(v).

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

2. If u is small, then for each $x \in \text{dist_all}(u)$, check temporal consistency and update $\text{dist_all}(v)$ accordingly.

After updating, the promotion condition is evaluated:

$$|\text{dist_all}(v)| \geq B.$$

If satisfied, vertex v is promoted to the large class, and its reachability table is discarded.

Algorithm 1 Processing a Temporal Edge

Require: Temporal edge (u, v, t)
Insert (u, t) into direct_edges(v)
if v is small **then**
  Update dist_all(v) with (u, t)
  if u is small **then**
    for each $(x, t_x) \in$ dist_all(u) **do**
      if $t_x \leq t$ **then**
        update (x, t) in dist_all(v)
      end if
    end for
  end if
end if
if $|\text{dist_all}(v)| \geq B$ **then**
  promote v to large
end if

3.5 Distance Query

For a query (s, t), two cases arise.

Case 1: t is small. The answer is obtained directly:

$$\text{return dist_all}(t)[s].$$

This requires $O(1)$ time.

Case 2: t is large. A backward traversal over incoming edges is performed:

- Only edges respecting temporal ordering are explored Casteigts et al. (2021); Bumpus & Meeks (2022).
- Upon reaching a small vertex, its precomputed reachability table is used.

Thus, large vertices are processed lazily, while small vertices provide constant-time lookups.

3.6 Hybrid Trade-off

The method combines two operational regimes:

Vertex type	Memory usage	Query time
Small	High	$O(1)$
Large	Low	Graph traversal

The threshold B controls the trade-off:

$$\text{memory} \longleftrightarrow \text{query latency}.$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

3.7 Structural Interpretation

The algorithm may be viewed as an adaptive reachability index:

- Locally sparse regions are collapsed into reachability tables.
- Globally dense or hub-like vertices remain explicit.

This adaptive separation aligns naturally with power-law degree distributions Barabási & Albert (1999); Clauset et al. (2009). In practice, large vertices tend to coincide with structural hubs, providing automatic detection of high-influence nodes without explicit preprocessing.

The approach can be interpreted as a hybridization of hub-based labeling Abraham et al. (2010), 2-hop labeling Cohen et al. (2003), and lazy exploration strategies Bumpus & Meeks (2022), while remaining fully adaptive to the input temporal graph.

4 Degree Distribution

We assume that the in-degree distribution of vertices follows a *power-law* (heavy-tailed) distribution Barabási & Albert (1999); Clauset et al. (2009):

$$\Pr[K \geq k] = \left(\frac{k}{k_{\min}}\right)^{1-\gamma}, \qquad \gamma > 1.$$

Such distributions are characteristic of many real-world networks, including social, communication, and transportation systems, where a small number of hub vertices possess very high degree, while the majority of vertices have low degree Kim & Anderson (2011).

In the context of temporal graphs, this heterogeneity explains the observed variability in reachable-set sizes:

- **Small vertices** with low degree typically have limited reachable sets and can efficiently store complete reachability tables;
- **Large vertices** with high degree act as structural hubs, strongly influencing reachability flow, and are more efficiently processed lazily to reduce memory usage Bumpus & Meeks (2022).

Therefore, the power-law degree distribution plays a central role in the construction of the hybrid index, as it enables analytical estimation of the fraction of small and large vertices and prediction of memory and query-time behavior.

5 Fraction of Large Vertices

Let K denote the in-degree of a vertex. Assume that the degree distribution has a power-law tail:

$$P(K = k) = Ck^{-\gamma}, \qquad k \geq k_{\min}, \quad \gamma > 1.$$

5.1 Normalization

The normalization constant C is determined by

$$\sum_{k=k_{\min}}^{\infty} Ck^{-\gamma} = 1.$$

Using a continuous approximation, we replace the sum with an integral:

$$\int_{k_{\min}}^{\infty} Ck^{-\gamma}\, dk = 1.$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Evaluating the integral,

$$C \int_{k_{\min}}^{\infty} k^{-\gamma}\, dk = C \left[\frac{k^{1-\gamma}}{1-\gamma} \right]_{k_{\min}}^{\infty}.$$

Since $\gamma > 1$, the upper limit vanishes, yielding

$$C \frac{k_{\min}^{1-\gamma}}{\gamma - 1} = 1.$$

Hence,

$$\boxed{C = (\gamma - 1) k_{\min}^{\gamma - 1}.}$$

5.2 Probability of Exceeding a Degree Threshold

The fraction of large vertices is defined as the probability that the degree exceeds a promotion threshold $k_p(B)$:

$$p_L(B) = \Pr[K \geq k_p(B)].$$

Using the continuous approximation,

$$p_L(B) = \int_{k_p(B)}^{\infty} C k^{-\gamma}\, dk.$$

Substituting the normalization constant,

$$p_L(B) = (\gamma - 1) k_{\min}^{\gamma - 1} \int_{k_p(B)}^{\infty} k^{-\gamma}\, dk.$$

Evaluating the integral,

$$\int_{k_p(B)}^{\infty} k^{-\gamma}\, dk = \frac{k_p(B)^{1-\gamma}}{\gamma - 1}.$$

After cancellation, we obtain

$$\boxed{p_L(B) = \left(\frac{k_p(B)}{k_{\min}} \right)^{1-\gamma}.}$$

5.3 Relation Between Degree Threshold and Promotion Parameter

Assume that the expected size of the reachable set of a vertex with degree k scales as a power function:

$$\mathbb{E}[R_{\text{in}}(k)] = C_R k^{\delta}.$$

The boundary between small and large vertices is defined by

$$C_R k_p^{\delta} = B.$$

Solving for $k_p(B)$ yields

$$k_p(B) = \left(\frac{B}{C_R} \right)^{1/\delta}.$$

5.4 Final Expression

Substituting $k_p(B)$ into the expression for $p_L(B)$, we obtain

$$p_L(B) = \left(\frac{(B/C_R)^{1/\delta}}{k_{\min}} \right)^{1-\gamma}.$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Rearranging terms,

$$p_L(B) = k_{\min}^{\gamma-1} C_R^{\frac{\gamma-1}{\delta}} B^{\frac{1-\gamma}{\delta}}.$$

Therefore,

$$\boxed{p_L(B) = A\, B^{\frac{1-\gamma}{\delta}}}$$

where

$$\boxed{A = k_{\min}^{\gamma-1} C_R^{\frac{\gamma-1}{\delta}}.}$$

5.5 Logarithmic Form

Taking logarithms,

$$\ln p_L = \frac{1-\gamma}{\delta} \ln B + \ln A.$$

Thus, the relationship is linear in log–log coordinates, with slope

$$\boxed{\alpha = \frac{1-\gamma}{\delta}.}$$

This power-law dependence enables empirical validation through log–log regression and provides a predictive model for how the fraction of large vertices decreases as the promotion threshold increases. (see Fig. 1).

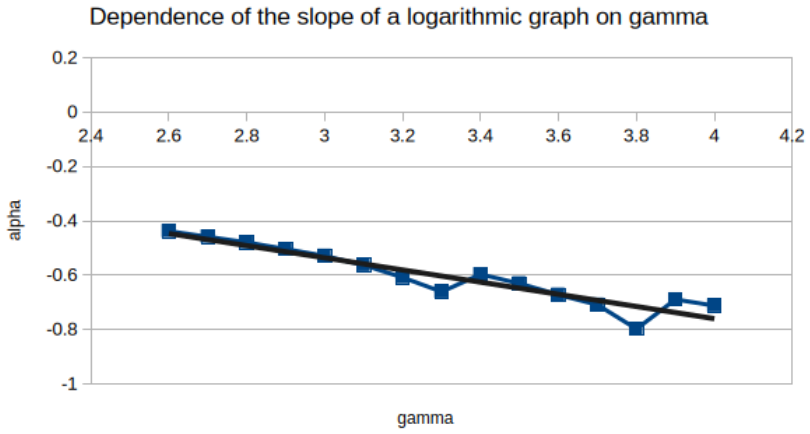


Figure 1: Dependence of the slope of a logarithmic graph on gamma

6 Memory Model

Let $n = |V|$ denote the number of vertices and $m = |E|$ the number of temporal edges (events).

Storing the underlying temporal graph requires

$$M_{\text{edges}} = O(m).$$

Additional memory is required to store reachability structures for small vertices.

Let $p_L(B)$ denote the fraction of large vertices under promotion threshold B. Then the fraction of small vertices equals $1 - p_L(B)$.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Each small vertex stores at most B reachable sources. Therefore, the total memory required for reachability structures is

$$M_{\text{reach}} = O\big((1 - p_L(B))\, n\, B\big).$$

Hence, the overall memory consumption is

$$\boxed{M(B) = O\big(m + (1 - p_L(B))\, n\, B\big).}$$

6.1 Asymptotic Form

Substituting the model for the fraction of large vertices

$$p_L(B) = A\, B^{\frac{1-\gamma}{\delta}},$$

we obtain

$$M(B) = O\Big(m + nB - nAB^{1+\frac{1-\gamma}{\delta}}\Big).$$

Let

$$\eta = 1 + \frac{1-\gamma}{\delta}.$$

Then

$$M(B) = O\big(m + nB - nAB^{\eta}\big).$$

6.2 Shape of the Memory Curve

The function $M(B)$ is the difference between two increasing terms:

- a linear growth term nB,
- a sublinear power term B^{η}.

Under typical heavy-tailed degree parameters,

$$0 < \eta < 1,$$

so for sufficiently large B, the linear term dominates:

$$M(B) \sim nB.$$

Thus, memory grows approximately linearly in the high-B regime.

Empirically, the observed curve exhibits a convex, "right-branch-of-a-parabola" shape: initially almost flat for small B, followed by nearly linear growth for larger thresholds (see Fig. 2).

6.3 Extreme Regimes

Small B. When the threshold is small, almost all vertices become large:

$$p_L(B) \approx 1.$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

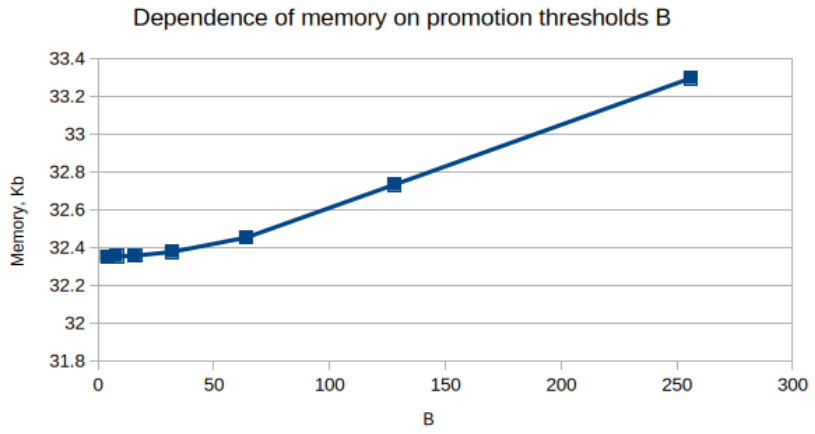


Figure 2: Dependence of memory on promotion thresholds B with $\gamma = 3.5$

Thus,

$$M(B) \approx O(m),$$

and memory becomes nearly independent of B.

Large B. When the threshold is large, most vertices are small:

$$p_L(B) \approx 0.$$

Then

$$M(B) \approx O(m + nB),$$

leading to approximately linear growth.

6.4 Practical Implication

Memory is monotonically increasing in B. Therefore, the optimal threshold must arise from a trade-off between memory growth and query-time behavior.

7 Query Time Model

We analyze the expected query response time as a function of the promotion threshold B.

Queries are processed differently depending on the vertex type:

- For **small** vertices, the answer is retrieved from the precomputed reachability table;
- For **large** vertices, a partial graph traversal is performed.

7.1 Baseline Model

Let

$$T_s = O(1), \qquad T_\ell = O(\tau),$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

where T_s denotes the average cost of a small-vertex query, and T_ℓ the average traversal cost for large vertices.

If $p_L(B)$ denotes the fraction of large vertices, then the expected query time equals

$$T(B) = (1 - p_L(B))T_s + p_L(B)T_\ell.$$

Substituting the power-law model

$$p_L(B) = A\,B^{\frac{1-\gamma}{\delta}},$$

we obtain

$$T(B) = T_s + (T_\ell - T_s)\,A\,B^{\frac{1-\gamma}{\delta}}.$$

Since $\gamma > 1$, the exponent is negative, and the baseline model predicts monotonic decay of query time as B increases.

7.2 Refined Model

Empirical results demonstrate instead a U-shaped dependence.(see Fig. 3).

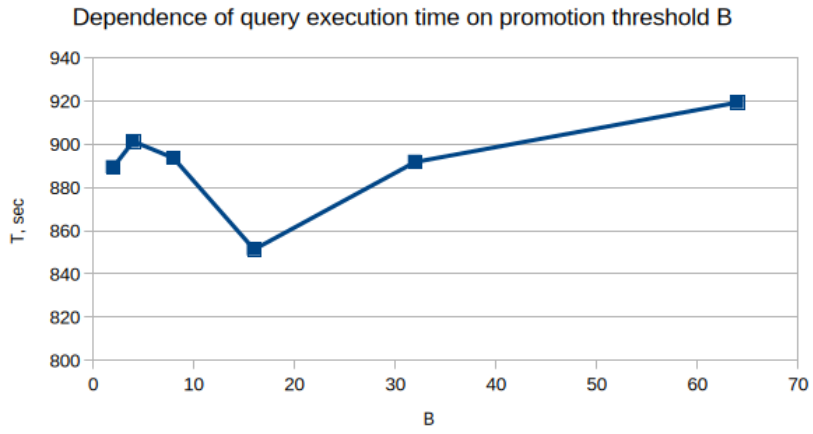


Figure 3: Dependence of query execution time on promotion threshold B

The discrepancy arises because T_s is not constant in practice. Reachability tables are implemented using hash tables (`unordered_map` and Robin Hood hashing), which provide $O(1)$ average complexity, but actual performance depends on:

- load factor,
- probe length,
- cache locality,
- memory hierarchy effects.

As B increases, table sizes scale approximately as nB, which degrades cache locality and increases average lookup latency.

Thus,

$$T_s = T_s(B), \qquad \frac{dT_s}{dB} > 0.$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

The refined model becomes

$$T(B) = (1 - p_L(B))\, T_s(B) + p_L(B)\, T_\ell(B).$$

8 Optimal Threshold

Empirical observations indicate that $T(B)$ exhibits a U-shaped dependence on B.

We model query time as

$$T(B) = aB^{-\beta} + cB^{\theta}, \qquad \beta > 0,\ \theta > 0,$$

where the first term captures reduction in large-vertex traversals, and the second term models increasing hash-table lookup cost.

Memory scales approximately as

$$M(B) \sim dB, \qquad d > 0.$$

8.1 Combined Cost Function

Define

$$C(B) = w_t T(B) + w_m M(B), \qquad w_t, w_m > 0.$$

Substituting,

$$C(B) = aB^{-\beta} + cB^{\theta} + dB.$$

8.2 Existence of a Minimum

We have

$$\lim_{B \to 0} C(B) = \infty, \qquad \lim_{B \to \infty} C(B) = \infty.$$

Thus, by continuity, a global minimum exists.

Under typical parameter values, the derivative

$$C'(B) = -a\beta B^{-\beta-1} + c\theta B^{\theta-1} + d$$

is strictly increasing, implying uniqueness of the minimizer.

9 Conclusion

We investigated a parameterized hybrid indexing algorithm for temporal reachability queries, controlled by a promotion threshold B.

Theoretical analysis showed that the fraction of large vertices decreases as a power law in B, while memory grows approximately linearly. A simplified model predicted monotonic improvement in query time.

However, empirical evaluation revealed a U-shaped behavior. We demonstrated that this phenomenon is explained by non-asymptotic effects of hash-table implementations, including cache locality degradation and memory hierarchy costs.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1014 DOI: 10.66693/mathai.1014
Special Issue: MathAI 2026 Selected Papers
Received: 13 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

As a result, an internal optimal threshold B^* exists, minimizing average query time.

The threshold parameter therefore acts as a tunable mechanism for balancing memory consumption and runtime performance in large-scale temporal graphs.

REFERENCES

Ittai Abraham, Daniel Delling, Andrew V. Goldberg, and Renato F. Werneck. A hub-based labeling algorithm for shortest paths on road networks. Technical Report MSR-TR-2010-165, Microsoft Research, 2010.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286 (5439):509–512, 1999.

Benjamin Merlin Bumpus and Kitty Meeks. Edge exploration of temporal graphs. *Theoretical Computer Science*, 913:1–18, 2022.

Arnaud Casteigts, Anne-Sophie Himmel, Hendrik Molter, and Philipp Zschoche. Finding temporal paths under waiting time constraints. *Algorithmica*, 83:2753–2799, 2021.

Arnaud Casteigts, Timothée Corsini, and Writika Sarkar. Simple, strict, proper, happy: A study of reachability in temporal graphs. *Theoretical Computer Science*, 2024.

Xiaoshuang Chen, Kai Wang, Xuemin Lin, Wenjie Zhang, Lu Qin, and Ying Zhang. Efficiently answering reachability and path queries on temporal bipartite graphs. *Proceedings of the VLDB Endowment*, 12(11):1473–1485, 2019.

Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.

Edith Cohen, Eran Halperin, Haim Kaplan, and Uri Zwick. Reachability and distance queries via 2-hop labels. *SIAM Journal on Computing*, 32(5):1338–1355, 2003.

Bolin Ding, Jeffrey Xu Yu, and Lu Qin. Finding time-dependent shortest paths over large graphs. *Proceedings of the VLDB Endowment*, 14(8):1413–1425, 2021.

Yong Ding, Xuemin Lin, and Jeffrey Xu Yu. Labels for reachability queries on very large graphs. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pp. 183–194, 2008.

Hyoungshick Kim and Ross Anderson. Temporal node centrality in complex networks. In *Proceedings of the 2011 IEEE International Conference on Communications (ICC)*, pp. 1–5, 2011.

Huanhuan Wu, James Cheng, Silu Huang, Yiping Ke, Yi Lu, and Yanyan Xu. Path problems in temporal graphs. *Proceedings of the VLDB Endowment*, 7(9):721–732, 2014.

Tianming Zhang, Yunjun Gao, Chen Lu, Wei Guo, and Shiliang Pu. Efficient distributed reachability querying of massive temporal graphs. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):866–879, 2019.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1015 DOI: 10.66693/mathai.1015
Special Issue: MathAI 2026 Selected Papers
Received: 15 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

A Systematic Study of Gate Functions in Soft Adaptive Policy Optimization

Egor Denisov[2], Svetlana Glazyrina[2], Maksim Kryzhanovskiy[2], Roman Ischenko[1,2]

[1]Institute for Artificial Intelligence, Lomonosov Moscow State University, Moscow, Russia
[2]Lomonosov Moscow State University, Moscow, Russia

Abstract

Group Relative Policy Optimization (GRPO) has significantly advanced the training of large language models and enhanced their reasoning capabilities, while it remains susceptible to instability due to the use of hard clipping. Soft Adaptive Policy Optimization (SAPO) addresses this limitation by replacing clipping with a smooth sigmoid-based gate function, which leads to more stable updates. We push this theory further and investigate the impact of different gate functions on both training stability and final model performance. We formalize the key properties that admissible gates should satisfy and propose several families of such functions for empirical evaluation. This paper presents an analysis of our findings based on experiments conducted with the Qwen2.5-7B-Instruct model on mathematical reasoning tasks. These results provide practical guidance for designing smoother and more robust policy optimization objectives for large language model training.

1 Introduction

Reinforcement learning (RL) has become a central tool for training large language models, enabling them to solve some of the most challenging problems across a wide range of different domains (Qwen-Team, 2025; DeepSeek-AI, 2025; Gemini-Team, 2025). The rapid progress in this field, together with the continuous growth in the scale of training data, necessitates the development of training algorithms that are both stable and highly scalable.

One of the most widely used RL algorithms in current practice is Group Relative Policy Optimization (GRPO) (Shao et al., 2024; DeepSeek-AI, 2026). GRPO can be considered as a development of Proximal Policy Optimization (PPO) (Schulman et al., 2017a) that eliminates the need for a separate value function network for advantage estimation. Instead, GRPO samples multiple outputs for the same query and computes group-based advantage estimates, leading to a reduction in computational demand and training cost. Similar to PPO, GRPO supports off-policy updates through the use of importance sampling, enabling the reuse of trajectories collected under previous policies and thereby improving sample efficiency. However, large deviations of the importance sampling ratio from unity may result in unstable training dynamics. To mitigate this issue, GRPO adopts the PPO clipping mechanism, which constrains policy updates and maintains proximity to the reference policy.

While clipping is practically effective, it cannot be regarded as a universally applicable method. Its hyperparameters are difficult to tune in order to balance training stability and exploration. A wide clipping range allows large importance ratio deviations to produce noisy and potentially destabilizing gradients, whereas a narrow clipping range suppresses most gradients, resulting in minimal policy updates, lack of exploration, and stagnated learning.

This limitation is addressed by Soft Adaptive Policy Optimization (SAPO) (Gao et al., 2025), which replaces hard clipping with a continuously differentiable gate function. The gradient of this function attains its maximum at an importance ratio of one and smoothly decays as the ratio moves away from this point. This design enables a gradual suppression of updates corresponding to extreme deviations from the reference policy, while preserving non-zero gradients and on-policy behavior. Additionally, SAPO introduces a temperature parameter that depends on the sign of the advantage, allowing the algorithm to more strongly encourage beneficial policy updates while being more conservative when penalizing unfavorable ones.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1015 DOI: 10.66693/mathai.1015
Special Issue: MathAI 2026 Selected Papers
Received: 15 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

The behavior of gate-based methods is largely determined by the effective width of the gradient peak around its maximum, as well as by the decay characteristics of the gradient for larger deviations of the argument. In the original SAPO paper, a sigmoid function is used, whose gradient decays exponentially. In this work, we explore alternative families of soft gates exhibiting diverse gradient decay behaviors, ranging from polynomial to Gaussian attenuation.

Our contributions are:

- We formalize the key properties that gate functions should satisfy and introduce several novel families of such functions.
- We propose a new metric, *Effective Update Ratio (EUR)*, which generalizes the *clip ratio* used in GRPO and provides a unified view of token-level policy updates.
- We conduct a comprehensive empirical study to assess how the choice of gate function affects training dynamics and downstream performance, evaluating our approach on mathematical reasoning benchmarks.

2 Related Works

Reinforcement learning has become a central paradigm for fine-tuning large language models beyond supervised approaches. Policy gradient methods typically rely on importance sampling to correct for distribution mismatch between the behavior and target policies. Trust Region Policy Optimization (TRPO) (Schulman et al., 2017b) enforces an explicit constraint on the Kullback–Leibler divergence between successive policies, while Proximal Policy Optimization (PPO) (Schulman et al., 2017a) addresses instability by introducing a clipped surrogate objective that restricts policy updates.

GRPO (Shao et al., 2024; DeepSeek-AI, 2026) was proposed as a scalable and value-free alternative to PPO, which replaces value function estimation with group-wise relative advantage normalization computed over multiple sampled outputs for the same prompt. Several extensions and variants of GRPO have been proposed to further improve stability and sample efficiency. GSPO (Zheng, 2025) implements sequence-level importance weights; GMPO (Zhao, 2025) modifies the aggregation of group statistics to reduce sensitivity to outliers; Dr. GRPO (Liu, 2025) removes normalization to avoid optimization bias; DAPO (Seed, 2025) applies dynamic sampling and decouples clipping bounds. Nevertheless, most GRPO-style methods retain PPO-inspired hard clipping and may inherit unstable updates and entropy collapse.

Several studies investigate alternatives to hard clipping with the aim of improving robustness (Wang et al., 2025; Sun et al., 2022; Su et al., 2025; Han et al., 2019; Garg et al., 2021; MiniMax, 2025). Soft Adaptive Policy Optimization (SAPO) (Gao et al., 2025), inspired by (Chen et al., 2022), leverages a smooth gate function that progressively down-weights samples with large importance ratios.

Despite the success of SAPO, there remains significant room for investigating the impact of an appropriate gate function on the method's performance.

3 Preliminaries

Group Relative Policy Optimization (GRPO). Let $\mathcal{Q}$ denote a query set ($\mathcal{Q} = \{q_i\}_{i=1}^{|\mathcal{Q}|}$). In GRPO (Shao et al., 2024; DeepSeek-AI, 2026) for each query $q \in \mathcal{Q}$, a group of G responses $\{o_1, \cdots, o_G\}$ from the behavior policy $\pi_{\theta_{old}}$ is sampled. The policy is being optimized by maximizing the following objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{Q},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(\cdot|q)} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\big(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_{i,t}\big) \right] \tag{1}$$

where $\{R_1, \cdots, R_G\}$ are rewards for each response, $r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}$ is the importance sampling ratio, $\hat{A}_{i,t} = \frac{R_i - \text{mean}(R)}{\text{std}(R)}$ is the normalized advantage of the *i*-th response, $\varepsilon > 0$ is the clipping threshold.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1015 DOI: 10.66693/mathai.1015
Special Issue: MathAI 2026 Selected Papers
Received: 15 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Soft Adaptive Policy Optimization (SAPO). SAPO (Gao et al., 2025) generalizes the idea of GRPO by introducing a smooth gate function instead of hard clipping. The objective transforms into the following:

$$\mathcal{J}_{SAPO}(\theta) = \mathbb{E}_{q\sim\mathcal{Q},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(.|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} f_{i,t}(r_{i,t}(\theta))\hat{A}_{i,t}\right], \tag{2}$$

where

$$f_{i,t}(x) = \sigma(\tau_{i,t}(x-1))\cdot\frac{4}{\tau_{i,t}}, \quad \sigma(x) = \frac{1}{1+e^{-x}}, \quad \tau_{i,t} = \begin{cases}\tau_{pos}, & \hat{A}_{i,t} > 0\\ \tau_{neg}, & \text{otherwise}\end{cases} \tag{3}$$

Gradient of the objective becomes:

$$\nabla_\theta \mathcal{J}_{SAPO}(\theta) = \mathbb{E}_{q\sim\mathcal{Q},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(.|q)} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} w_{i,t}(\theta) r_{i,t}(\theta)\nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})\hat{A}_{i,t}\right], \tag{4}$$

where

$$w_{i,t}(\theta) = \left.\frac{\mathrm{d}f_{i,t}}{\mathrm{d}x}\right|_{x=r_{i,t}(\theta)} = \tau f_{i,t}(r_{i,t}(\theta))(1-\frac{\tau}{4}f_{i,t}(r_{i,t}(\theta))) \tag{5}$$

$w_{i,t}(\theta)$ reaches its maximum equal to 1 at $r_{i,t}(\theta) = 1$, which corresponds to the on-policy behavior. As the importance ratio deviates from 1, the gradient starts decreasing exponentially towards 0. This allows more stable training without limiting model exploration.

4 METHODOLOGY

We hypothesize that an appropriate choice of the gate function in the SAPO algorithm can lead to improved convergence during model training by enabling an optimal contribution of individual tokens in the optimization process. Before exploring and analyzing alternatives, it is necessary to formalize a set of properties that all candidate functions $f : \mathbb{R}_+ \to \mathbb{R}$ should satisfy: (i) the function should be continuously differentiable; (ii) the derivative $f'(x)$ should attain its global maximum that equals to 1 at $x = 1$; (iii) the derivative should decrease monotonically as the argument moves away from 1; and (iv) $f'(x)\cdot x \to 0$ as $x \to \infty$.

The suggested properties admit a clear optimization interpretation. (ii) ensures that when $r_{i,t}(\theta) = 1$, the resulting gradient contribution coincides with the on-policy update, thereby preserving consistency with standard policy gradient optimization. (iii) enforces a controlled attenuation of samples whose importance ratios deviate from 1. This behavior stabilizes training by discouraging overly aggressive updates induced by off-policy samples. Finally, (iv) guarantees that samples associated with extreme importance ratios contribute negligibly to the gradient, effectively suppressing the impact of outliers and preventing instability caused by heavy-tailed ratio distributions.

We consider the sigmoid applied in the original paper as a baseline and propose the following alternatives for the gate function: Error function (Normal CDF) (6), Arctangent (7) and Softsign (8). Consistent with the original paper, we incorporate a temperature parameter τ into the objective. The additive constants introduced in the gate functions do not affect their gradients and are included solely for interpretability. Specifically, these constants ensure that the resulting gate functions remain positive on the half-interval $[0; +\infty)$ and are normalized to pass through the point $(1, 1)$.

$$f_{\mathrm{erf}}(x) = \sqrt{\frac{\pi}{2\tau^2}}\left(1+\mathrm{erf}\left(\frac{\tau(x-1)}{\sqrt{2}}\right)\right) + 1 - \sqrt{\frac{\pi}{2\tau^2}}, \quad f'_{\mathrm{erf}}(x) = \exp\left(-\frac{\tau^2(x-1)^2}{2}\right) \tag{6}$$

$$f_{\arctan}(x) = 1 + \frac{1}{\tau}\arctan(\tau(x-1)), \quad f'_{\arctan}(x) = \frac{1}{(1+\tau^2(x-1)^2)} \tag{7}$$

$$f_{\mathrm{softsign}}(x) = 1 + \frac{x-1}{\sqrt{1+\tau^2(x-1)^2}}, \quad f'_{\mathrm{softsign}}(x) = \frac{1}{(1+\tau^2(x-1)^2)^{3/2}} \tag{8}$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1015 DOI: 10.66693/mathai.1015
Special Issue: MathAI 2026 Selected Papers
Received: 15 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

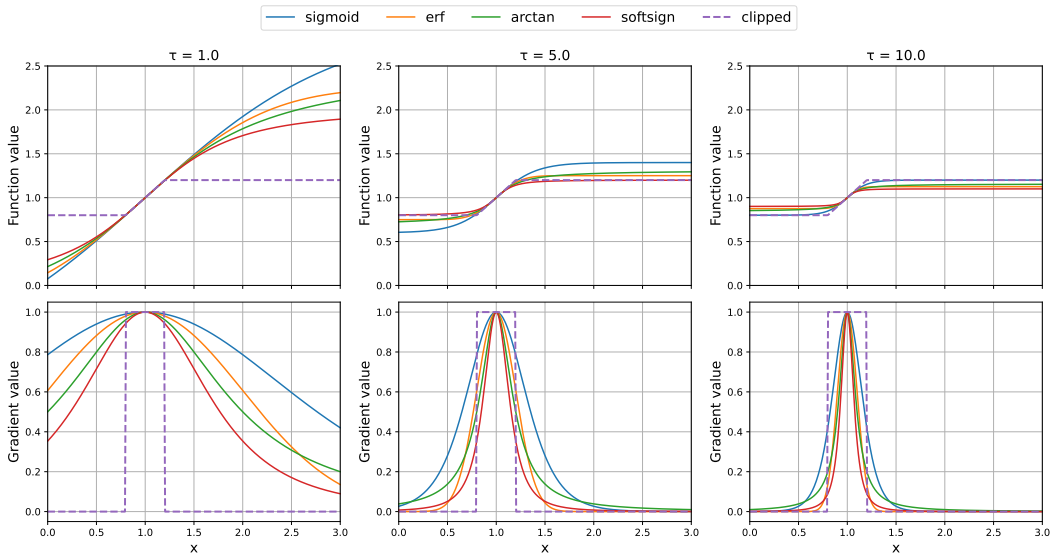


Figure 1: Temperature-dependent behavior of the considered gate functions (top row) and their gradients (bottom row) for $\tau \in \{1, 5, 10\}$. All functions are normalized to be positive on $[0; +\infty)$ and to pass through the point $(1, 1)$. Increasing the temperature sharpens the transition around $x = 1$, leading to more localized and higher gradient peaks for smooth gates, while the clipped variant exhibits piecewise-linear behavior with constant gradients in its active region.

The behavior of the selected functions, in comparison with hard clipping and the sigmoid function used in the original SAPO formulation, is illustrated in Figure 1.

While all considered functions satisfy the aforementioned properties, they differ substantially in the behavior of their derivatives. The derivatives of arctangent and softsign exhibit polynomial decay, and the derivative of the error function follows a Gaussian decay. Heavier-tailed gradients amplify the influence of rare tokens on the learning process, thereby increasing exploration, albeit potentially at the cost of reduced stability. Conversely, lighter-tailed gradients of f_{erf} make the method increasingly similar to hard clipping by progressively suppressing the contribution of extreme tokens.

5 Experiments

5.1 Experimental setup

All experiments were conducted using the Qwen2.5-7B-Instruct model on mathematical reasoning tasks.

Training. The training data consisted of an equal mixture of the GSM8K (train split) (Cobbe et al., 2021) and DeepMath(He et al., 2025) datasets, and the model was aligned via a cold-start reinforcement learning procedure without supervised warm-up. For each prompt, we generated 8 rollouts with a maximum response length of 512 tokens. The per-device batch size was set to 1, and gradients were accumulated over 16 steps, yielding a larger effective batch size. To improve sample efficiency and to better isolate the effect of gate function alteration, two gradient updates were performed for each batch, enabling reuse of previously generated trajectories. Training was performed on 8 NVIDIA A100 GPUs for a total of 5,000 optimization steps. Generation parameters are provided at Appendix A.

Reward. The reward function was defined as a sum of an answer correctness component and a formatting component:

$$r = r_{\text{answer}} + r_{\text{format}}. \tag{9}$$

r_{format} evaluates whether the model follows a prescribed structured response template of the form *<think> text </think> <answer> text </answer>*, where *<think>, </think>, <answer>* and

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1015 DOI: 10.66693/mathai.1015
Special Issue: MathAI 2026 Selected Papers
Received: 15 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

</*answer*> denote special tokens and *text* represents arbitrary generated content:

$$r_{\text{format}} = \begin{cases} 1, & \text{format is fully satisfied} \\ 0.5, & \text{generation begins with the <}\textit{think}\text{> and ends with the </}\textit{answer}\text{>} \\ 0.25, & \text{generation begins with the <}\textit{think}\text{> or ends with the </}\textit{answer}\text{>} \\ 0, & \text{otherwise} \end{cases} . \quad (10)$$

r_{answer} measures solution correctness. We extract the text generated by the model between the <*answer*> and </*answer*> tokens and evaluate it for consistency with the ground-truth solution.

$$r_{\text{answer}} = \begin{cases} 1, & \text{model output matches the correct answer} \\ 0, & \text{otherwise} \end{cases} . \quad (11)$$

Evaluation. The trained models were evaluated on several mathematical benchmarks of varying difficulty: GSM8K (test split), MATH500 and AIME. GSM8K (Cobbe et al., 2021) consists of grade school arithmetic word problems requiring multi-step reasoning. MATH500 is a curated subset of the MATH (Hendrycks et al., 2021) dataset containing competition-level problems across algebra, geometry, number theory, and combinatorics. AIME comprises problems from the corresponding editions of the American Invitational Mathematics Examination, designed to assess advanced mathematical reasoning and precise numerical answer generation. During evaluation, responses were generated with a sampling temperature of 0.7. We used the accuracy metric for measuring model performance.

5.2 Results

Rewards. We explore several temperature settings for the proposed gate functions. As shown in Figure 2, for the erf-based gate temperature hyperparameter pair $\tau_{pos} = 10, \tau_{neg} = 12$ yields the best results. These values provide an optimal trade-off between aggressive clipping ($\tau_{pos} = 13, \tau_{neg} = 15$) and fully accounting for the contribution of all tokens ($\tau_{pos} = 1, \tau_{neg} = 1.05$). Similar behavior was observed across all considered gate functions including sigmoid from SAPO. Therefore, in the subsequent comparative analysis, we report results corresponding to this temperature configuration.

Figure 3 illustrates reward dynamics for the considered methods in comparison with original SAPO and GRPO with $\varepsilon = 0.2$. At the early stages of training, all SAPO-based variants demonstrate steeper increase in rewards. Leveraging soft gates leads to faster exploration and facilitates 4 times quicker acquisition of the correct response format. Overall, the use of the error function and the arctangent gating mechanisms results in superior performance.

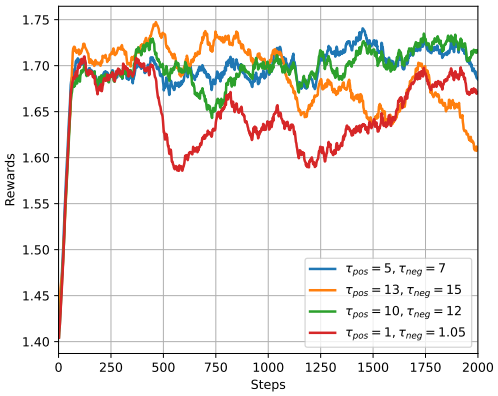


Figure 2: Comparison of reward dynamics across training steps for f_{erf} with various temperature configurations.

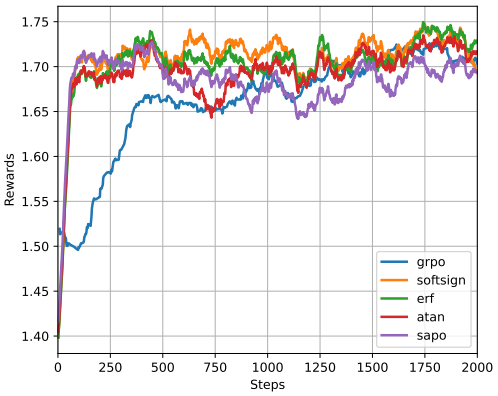


Figure 3: Comparison of reward dynamics across training steps for all considered methods with best configurations. For GRPO $\varepsilon = 0.2$, for SAPO-like methods $\tau_{pos} = 10, \tau_{neg} = 12$.

Entropy and EUR. In addition, we investigate the behavior of policy entropy depending on both the chosen method and the temperature parameter. We introduce a quantitative indicator termed

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1015 DOI: 10.66693/mathai.1015
Special Issue: MathAI 2026 Selected Papers
Received: 15 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

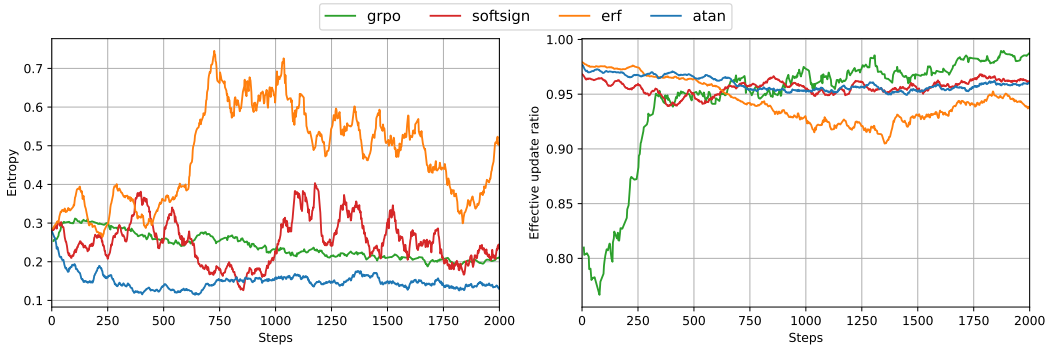


Figure 4: Policy entropy (left) and effective update ratio (right) over training for different gate functions and GRPO. For all SAPO-like methods τ_{pos} is set to 10 and τ_{neg} is set to 12.

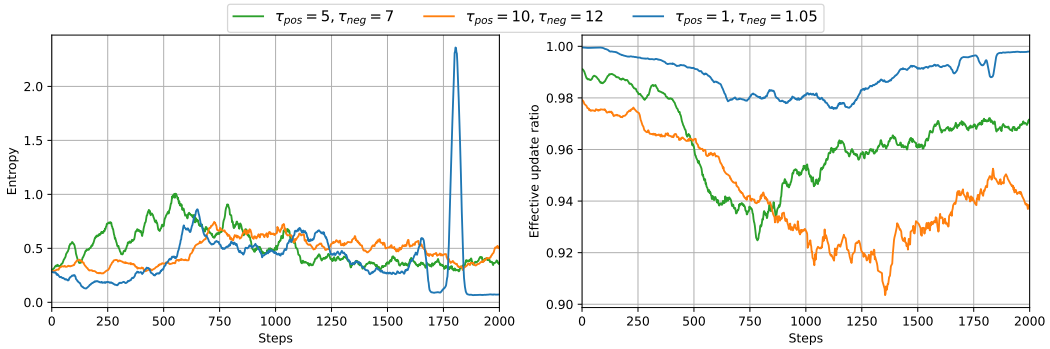


Figure 5: Policy entropy (left) and effective update ratio (right) over training for f_{erf} with different temperature configurations.

Effective Update Ratio (EUR). EUR is defined as the derivative of the gate function evaluated at $r_{i,t}(\theta)$ averaged over all tokens used at the current training step:

$$\text{EUR} = \frac{1}{B}\sum_{i=1}^{B}\frac{1}{\sum_{j=1}^{G}|o_j|}\sum_{j=1}^{G}\sum_{k=1}^{|o_j|} f'_{j,k}(r_{j,k}(\theta)) \tag{12}$$

where B represents batch size, G is the number of outputs for each prompt. For GRPO we have $f'_{j,k}(r_{j,k}(\theta)) = \begin{cases} \mathbf{1}[r_{j,k}(\theta) \le 1+\varepsilon], & \hat{A}_j > 0 \\ \mathbf{1}[r_{j,k}(\theta) \ge 1-\varepsilon], & \hat{A}_j \le 0 \end{cases}$, where $\mathbf{1}[.]$ denotes the indicator function. Under this definition, the EUR for GRPO reduces to $1 - $ *clip_ratio*, where *clip_ratio* is a fraction of tokens with importance sampling ratio out of clipping range. EUR takes values from $[0, 1]$ and can be interpreted as the mean fraction of token contribution to the parameter update at a given training step.

As observed from Figure 4, an increase in policy entropy is associated with a decrease in EUR. This effect can be explained by the higher probability of sampling tokens with large importance ratios that significantly deviate from 1. Such tokens contribute less to the policy update and, consequently, reduce the Effective Update Ratio. Thus gates still act as a form of a soft policy constraint to prevent destabilizing updates and the following memory loss.

Figure 5 illustrates the dependence of policy entropy and EUR for f_{erf} on the temperature parameter. We discover that increasing the temperature leads to a decrease in EUR, as the gradient becomes more sharply peaked around 1 and decays more rapidly away from this point. At lower temperatures ($\tau = 5$), the entropy reaches higher values (for $\tau = 1$ training becomes unstable and collapses). Nevertheless, even a high value of temperature parameter is sufficient to prevent the entropy decay observed in GRPO.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1015 DOI: 10.66693/mathai.1015
Special Issue: MathAI 2026 Selected Papers
Received: 15 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Benchmark results. Table 1 summarizes results of best configurations for each gate function to assess their empirical performance on mathematical tasks. The best results on each benchmark are highlighted in bold.

Model	GSM8K	MATH500	AIME
Qwen2.5-7B-Instruct	80.3	56.6	3.3
+ GRPO	80.6	**58.6**	8.3
+ SAPO (sigmoid baseline)	82.2	57.8	8.3
+ SAPO (erf)	82.7	56.2	**10.0**
+ SAPO (arctan)	82.3	55.8	5.0
+ SAPO (softsign)	**82.8**	57.2	5.0

Table 1: Accuracy (%) of different model configurations on mathematical reasoning benchmarks.

The methods we proposed achieve improvements on most benchmarks (82.8% for f_{softsign} on GSM8K and 10% for f_{erf} on AIME). To fully realize their potential, a more careful selection of training data could be considered, which would enable the model to perform more extensive reasoning.

6 Conclusion

We formalized the key properties that gate functions must satisfy to ensure stable and effective policy updates. We introduced several novel gate function variants for SAPO and systematically investigated their impact on the method's behavior. In addition, we analyzed these approaches through the lens of entropy dynamics and a newly proposed metric, the Effective Update Ratio, which quantifies the relative contribution of tokens to the policy update at each training step. Our empirical results indicate that the erf-based gate shows the strongest gains on AIME and improves GSM8K vs. sigmoid baseline, while MATH500 remains challenging. Furthermore, we observe that overall performance is highly sensitive to the choice of temperature parameters, which must strike a careful balance between overly aggressive clipping and excessively smooth updates that incorporate nearly all tokens. This sensitivity suggests that principled temperature selection and adaptive scheduling strategies constitute a promising direction for future research.

Computational Resources

The research was carried out using the MSU-270 supercomputer of Lomonosov Moscow State University.

References

Xing Chen et al. The sufficiency of off-policyness and soft clipping: Ppo is still insufficient according to an off-policy measure. *arXiv:2205.10047v6*, 2022.

Karl Cobbe et al. Training verifiers to solve math word problems. *arXiv:2110.14168v2*, 2021.

DeepSeek-AI. Deepseek-v3 technical report. *arXiv:2412.19437v2*, 2025.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv:2501.12948v2*, 2026.

Chang Gao et al. Soft adaptive policy optimization. *arXiv:2511.20347*, 2025.

Saurabh Garg et al. On proximal policy optimization's heavy-tailed gradients. *arXiv:2102.10264v2*, 2021.

Gemini-Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv:2507.06261v6*, 2025.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1015 DOI: 10.66693/mathai.1015
Special Issue: MathAI 2026 Selected Papers
Received: 15 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Seungyul Han et al. Dimension-wise importance sampling weight clipping for sample-efficient reinforcement learning. *arXiv:1905.02363v2*, 2019.

Zhiwei He et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv:2504.11456v2*, 2025.

Dan Hendrycks et al. Measuring mathematical problem solving with the math dataset. *arXiv:2103.03874v2*, 2021.

Zichen Liu. Understanding r1-zero-like training: A critical perspective. *arXiv:2503.20783v2*, 2025.

MiniMax. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv:2506.13585v1*, 2025.

Qwen-Team. Qwen3 technical report. *arXiv:2505.09388*, 2025.

John Schulman et al. Proximal policy optimization algorithms. *arXiv:1707.06347v2*, 2017a.

John Schulman et al. Trust region policy optimization. *arXiv:1502.05477v5*, 2017b.

ByteDance Seed. Dapo: An open-source llm reinforcement learning system at scale. *arXiv:2503.14476v2*, 2025.

Zhihong Shao et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300v3*, 2024.

Zhenpeng Su et al. Ce-gppo: Coordinating entropy via gradient-preserving clipping policy optimization in reinforcement learning. *arXiv:2509.20712v4*, 2025.

Mingfei Sun et al. You may not need ratio clipping in ppo. *arXiv:2202.00079v1*, 2022.

Jiakang Wang et al. Aspo: Asymmetric importance sampling policy optimization. *arXiv:2510.06062v1*, 2025.

Yuzhong Zhao. Geometric-mean policy optimization. *arXiv:2507.20673v3*, 2025.

Chujie Zheng. Group sequence policy optimization. *arXiv:2507.18071v2*, 2025.

APPENDICES

A TRAINING PARAMETERS

Table 2 summarizes the training and generation hyperparameters used in our experiments. All configurations were kept fixed across runs unless explicitly stated otherwise.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1015 DOI: 10.66693/mathai.1015
Special Issue: MathAI 2026 Selected Papers
Received: 15 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Category	**Parameter**	**Value**
Generation	MAX_PROMPT_LENGTH	1536
	MAX_COMPLETION_LENGTH	512
	NUM_GENERATIONS	8
Batching	PER_DEVICE_TRAIN_BATCH_SIZE	1
	GRAD_ACCUM	16
	NUM_ITERATIONS	2
Precision	BF_16	TRUE
Optimization	OPTIMIZER	AdamW
	LEARNING_RATE	10^{-6}
	WARMUP_STEPS	0
	SCHEDULER_TYPE	linear
AdamW Parameters	ADAM_BETA1	0.9
	ADAM_BETA2	0.999
	ADAM_EPSILON	10^{-8}

Table 2: Training and generation hyperparameters used across all experiments.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

IMPROVING FAIRNESS IN AI-POWERED RECRUITMENT: AN INTERPRETABLE RESUME SCREENING SYSTEM

Natalia A. Agapova
Innopolis University
Innopolis, Russia
n.agapova@innopolis.university

Rustam A. Lukmanov
Research Center of Artificial Intelligence
Innopolis University
Innopolis, Russia
r.lukmanov@innopolis.university

ABSTRACT

Modern automated resume screening systems are typically based on neural text classification models that encode a resume as a feature representation and predict a discrete label corresponding to candidate category, suitability level, or job role. Such models commonly produce class logits parameterized by model weights, which are converted into class probabilities via the softmax function over the target classes. These models are typically trained using cross-entropy loss and deployed as the first stage of automated candidate filtering.

Despite their effectiveness, resume classifiers may encode implicit bias through correlations between predictions and non-job-related or proxy textual features. To study this effect, we analyze feature influence using Integrated Gradients, which assign an attribution score to each input feature based on the path integral of partial derivatives between a baseline representation and the actual input. This analysis reveals systematic dependencies on features that should be irrelevant to candidate evaluation.

Building on these observations, we evaluate multiple debiasing techniques and propose an interpretability-guided framework for bias mitigation. We compare six methods spanning in-processing approaches (GroupDRO, Focal Loss, Label Smoothing, Adversarial debiasing) and attribution-based approaches (Data Scrubbing, Attention Regularization) that leverage the interpretability findings directly. This allows explainable analysis to guide the development of fairer resume screening models.

1 INTRODUCTION

The adoption of machine learning in Human Resources (HR) has grown rapidly, with automated systems now screening millions of resumes annually (Raghavan et al., 2019). While these systems promise efficiency and consistency, they risk encoding and amplifying societal biases present in historical hiring data.

Bias in automated hiring systems can manifest through multiple protected attributes: gender, age, ethnicity, socioeconomic background, and geographic location. Such biases lead to systematic disadvantages for certain candidate groups—not due to their qualifications, but due to spurious correlations learned by the model. For instance, models may learn to associate certain vocabulary patterns, institutional names, or regional expressions with hiring outcomes, even when these features carry no legitimate signal about candidate suitability.

In this paper, we present a comprehensive framework for analyzing and mitigating bias in BERT-based resume classification models. We examine multiple potential sources of bias, including geographic location, gender, and age. Our analysis reveals that **geographic bias** (across city groups) exhibits the strongest measurable disparities in our dataset, making it the primary focus of our debiasing efforts. However, the proposed methodology generalizes to any categorical sensitive attribute.

Our contributions are:

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

1. **Multi-attribute Bias Analysis**: We analyze bias across geographic, gender, and age attributes using True Positive Rate (TPR) gaps, identifying city groups as the dominant source of measurable disparities.
2. **Robust Fairness Metrics**: We introduce a filtering methodology that excludes statistically unreliable subgroups (support < 30), revealing that naive fairness metrics can substantially overestimate bias due to small-sample noise.
3. **Interpretability-Driven Bias Detection**: We apply Integrated Gradients to the baseline model to identify specific tokens (city names, age-related terms) that systematically influence predictions, providing evidence of encoded bias before any mitigation.
4. **Comparative Debiasing Study**: We systematically evaluate six debiasing techniques spanning in-processing methods (GroupDRO, Focal Loss, Label Smoothing, Adversarial) and attribution-based methods (Data Scrubbing, Attention Regularization).

2 Related Work

2.1 Fairness in Machine Learning

Algorithmic fairness has emerged as a critical research area, with multiple competing definitions proposed. Dwork et al. (2012) introduced *fairness through awareness*, arguing that similar individuals should receive similar outcomes. Hardt et al. (2016) formalized *equalized odds*, requiring equal true positive and false positive rates across demographic groups. Chouldechova (2017) demonstrated inherent tensions between calibration and equalized odds, showing that satisfying both simultaneously is often impossible. Caton & Haas (2024) provide a comprehensive survey of fairness metrics and their trade-offs.

2.2 Bias in HR and Recruitment Systems

Automated hiring systems present unique fairness challenges. Raghavan et al. (2019) evaluated commercial resume screening tools and found significant disparities in outcomes across demographic groups. Köchling & Wehner (2020) conducted a systematic review of algorithmic discrimination in HR, identifying data bias, algorithmic bias, and deployment bias as key sources. Chen (2023) examined ethical implications of AI-enabled recruitment, highlighting the tension between efficiency gains and fairness concerns. Deshpande et al. (2020) addressed demographic bias in resume filtering using adversarial debiasing, focusing primarily on gender attributes.

2.3 Debiasing Methods

Bias mitigation techniques span three stages: pre-processing, in-processing, and post-processing (Hort et al., 2023). Pre-processing methods modify training data through reweighting or resampling. In-processing methods incorporate fairness constraints into the learning objective. Hashimoto et al. (2018) proposed Group Distributionally Robust Optimization (GroupDRO), which optimizes for worst-case group performance:

$$\min_{\theta} \max_{g \in \mathcal{G}} \mathbb{E}_{(x,y)\sim P_g}[\mathcal{L}(f_\theta(x), y)] \tag{1}$$

where $\mathcal{G}$ represents demographic groups. Zhang et al. (2018) introduced adversarial debiasing, training a classifier to be uninformative about protected attributes.

2.4 Interpretability for Fairness

Explainable AI (XAI) methods can reveal sources of bias in model predictions. Sundararajan et al. (2017) introduced Integrated Gradients, providing axiomatic guarantees for feature attribution. We use this technique to identify tokens contributing to potentially biased predictions, enabling both diagnosis and targeted mitigation.

Gap in literature. While prior work has addressed gender and racial bias in resume screening, comprehensive multi-attribute analysis with robust evaluation metrics remains limited, particularly in non-English contexts, and a definitive solution to model bias remains elusive.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

3 Problem Setup

3.1 Task Definition

We formulate resume classification as a multi-class classification problem. Given a resume text $x \in \mathcal{X}$, we predict the professional category $y \in \mathcal{Y} = \{1, \ldots, K\}$ where $K = 9$ categories. Each resume is associated with multiple sensitive attributes: geographic location $g_{city} \in \mathcal{G}_{city}$, gender $g_{gender} \in \{\text{Male}, \text{Female}\}$, and age group $g_{age} \in \mathcal{G}_{age}$.

3.2 Dataset

We use a dataset of resumes from the leading job portal in Russia. The dataset comprises 27,550 samples split into training (16,530), validation (5,510), and test (5,510) sets. Resumes span 41 city groups and 7 age categories.

The 9 professional categories are: *backend_general_dev*, *web_frontend*, *sysadmin_devops_network*, *project_product*, *sales_account*, *tech_support_helpdesk*, *it_governance_leadership*, *technical_specialized*, and *generic_it_ops*.

3.3 Attribute Distribution Analysis

Before selecting which sensitive attribute to focus on for debiasing, we analyzed the distribution of each attribute:

- **City groups**: 41 unique cities with severe imbalance. Moscow and Saint Petersburg comprise over 40% of samples, while many cities have fewer than 50 samples. This creates substantial subgroup variance.
- **Gender**: Binary distribution (Male/Female) with approximately 70%/30% split. More balanced but fewer distinct groups.
- **Age**: 7 groups (<18, 18–21, 22–25, 26–35, 36–50, 50+, Unknown). Moderate imbalance with concentration in 26–35 range.

Preliminary TPR gap analysis revealed that **city groups exhibit the largest measurable disparities** (robust worst-case gap of 32.9% vs. 18.2% for age and 12.4% for gender). We therefore focus our debiasing experiments on city groups while demonstrating the generalizability of our methodology.

4 Methodology

4.1 Baseline Model

We fine-tune BERT-base-uncased (Devlin et al., 2018) for sequence classification. Input resumes are tokenized with maximum length 128 and classified using a linear head:

$$\hat{y} = \text{softmax}(W \cdot h_{[CLS]} + b) \tag{2}$$

where $h_{[CLS]} \in \mathbb{R}^{768}$ is the BERT [CLS] token representation.

To address severe group imbalance (Section 3.3), we apply inverse square-root sample reweighting during training:

$$w_i = \frac{1}{\sqrt{n_{g_i}}} \cdot \frac{1}{Z} \tag{3}$$

where n_{g_i} is the number of training samples in group g_i and Z is a normalization constant. This gives higher weight to underrepresented groups while avoiding extreme weights that could destabilize training.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

4.2 Fairness Metrics

We evaluate fairness using True Positive Rate (TPR) gaps across demographic groups. For a class c and group g:

$$\text{TPR}_{c,g} = P(\hat{y} = c | y = c, G = g) \tag{4}$$

The TPR gap measures disparity:

$$\Delta\text{TPR}_c = \max_{g \in \mathcal{G}} \text{TPR}_{c,g} - \min_{g \in \mathcal{G}} \text{TPR}_{c,g} \tag{5}$$

We report **worst-case** gap ($\max_c \Delta\text{TPR}_c$) and **macro-average** gap.

4.3 Robust Fairness Evaluation

Small subgroups produce unreliable TPR estimates. We filter group-class combinations with support below $\tau = 30$:

$$\Delta\text{TPR}_c^{\text{robust}} = \max_{g: n_{c,g} \geq \tau} \text{TPR}_{c,g} - \min_{g: n_{c,g} \geq \tau} \text{TPR}_{c,g} \tag{6}$$

4.4 Bias Detection via Integrated Gradients

Before applying debiasing methods, we first verify that the baseline model exhibits bias by analyzing which input features influence its predictions. We use Integrated Gradients (Sundararajan et al., 2017), which assigns an attribution score to each input token:

$$\text{IG}_i(x) = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha(x - x'))}{\partial x_i} d\alpha \tag{7}$$

where x_i is the embedding of the i-th token, x' is a baseline (zero embedding), and F is the model's output logit for the predicted class. This method satisfies key axioms including completeness (attributions sum to the prediction difference) and sensitivity (relevant features receive non-zero attribution).

We compute attributions for representative samples across demographic subgroups, selecting examples that illustrate both correct classifications and systematic errors, and analyze tokens related to sensitive attributes: city names and age-related terms.

4.4.1 Geographic Bias in Baseline Model

Figure 1 shows the mean attribution scores for city name tokens across professional categories. The results reveal systematic geographic bias:

- **Moscow** has strong *negative* attribution (-0.164) for *backend_general_dev*, suggesting the model associates Moscow-based candidates with other professions.
- **Saint Petersburg** shows strong *positive* attribution ($+0.159$) for *sales_account*, indicating geographic stereotyping.
- Regional cities (Ekaterinburg, Novosibirsk, Chelyabinsk) consistently receive positive attribution for *sysadmin_devops_network* ($+0.038$ to $+0.082$).

These patterns demonstrate that the model uses geographic information—which should be irrelevant to professional qualification—as a predictive signal.

Our analysis flagged multiple bias indicators: the token "москва" (Moscow) received inconsistent attributions across samples—negative (-0.164) for age 22–25 errors but positive ($+0.197$) for backend misclassifications. Similarly, regional cities like "магнитогорск" (Magnitogorsk) and "томская" (Tomsk) showed systematic positive attribution, confirming geographic signal leakage.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

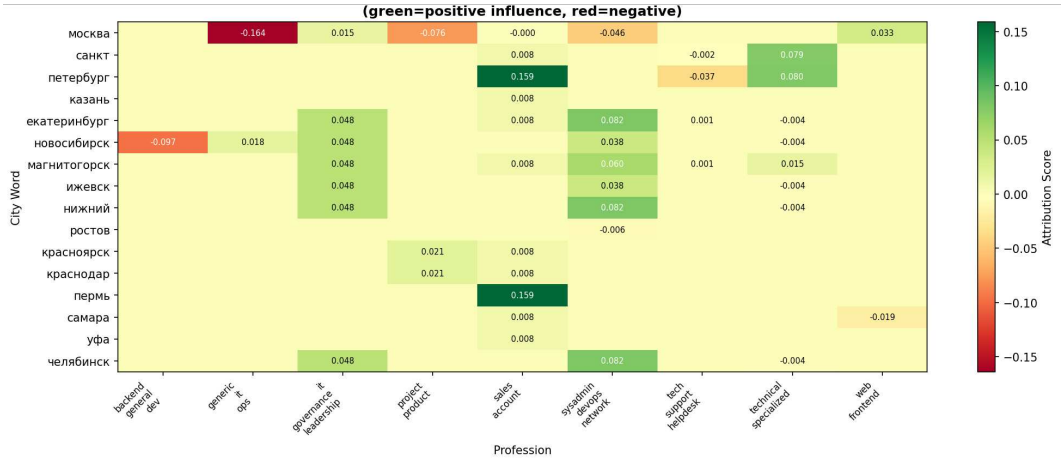


Figure 1: Integrated Gradients attribution scores for city name tokens across professional categories in the baseline model. Green indicates positive influence toward the profession; red indicates negative influence. The model exhibits systematic geographic bias, using city names as predictive features.

4.4.2 Age-Related Bias in Baseline Model

Figure 2 presents attribution scores for age-related vocabulary. Key findings:

- The token "пенсионер" (pensioner) has a strong negative attribution (-0.097) for *backend_general_dev* but positive attribution ($+0.074$) for *it_governance_leadership*, reflecting age stereotypes about technical vs. managerial roles.
- The token "студент" (student) receives positive attribution ($+0.092$) for *sysadmin_devops_network* but negative attribution (-0.014) for *tech_support_helpdesk*.
- Age numbers ("50", "60") show differential patterns: older ages correlate negatively with technical roles and positively with governance positions.

These interpretability findings confirm that the baseline model learns spurious correlations between protected attributes and professional categories, motivating the debiasing methods described below.

We did not construct a dedicated gender attribution heatmap because no explicit gender-indicative tokens were identified in the vocabulary; the aggregate gender-level attributions shown in Appendix A (Figure 4) confirm that gender signal is diffuse rather than concentrated in specific tokens. Gender bias may be encoded through indirect proxies (surnames, job titles) not easily captured by token-level analysis.

Additional per-sample attribution examples are provided in Appendix A.

4.5 Debiasing Methods

Based on the bias patterns identified through Integrated Gradients analysis, we evaluate six debiasing techniques:

4.5.1 GroupDRO

Optimizes for worst-case group performance with dynamic group weights:

$$q_g^{(t+1)} \propto q_g^{(t)} \cdot \exp(\eta \cdot \mathcal{L}_g^{(t)}) \tag{8}$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

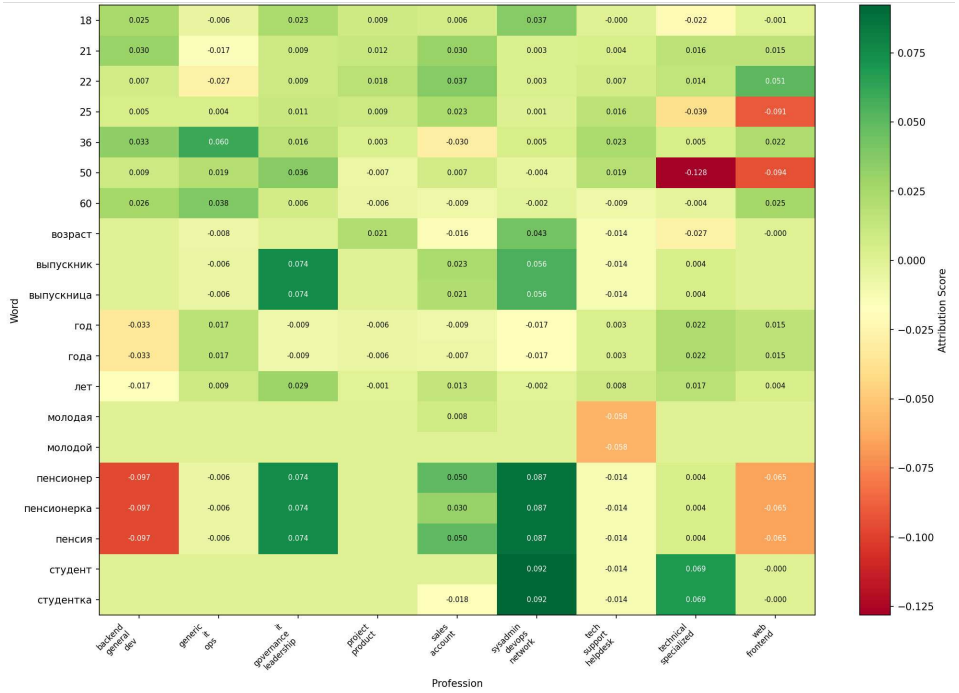


Figure 2: Attribution scores for age-related words across professions. The baseline model encodes age stereotypes: "pensioner" negatively influences backend development predictions while positively influencing IT governance.

4.5.2 Focal Loss

Down-weights well-classified examples:

$$\mathcal{L}_{FL} = -\alpha_t(1-p_t)^\gamma \log(p_t) \tag{9}$$

4.5.3 Label Smoothing

Distributes probability mass to non-target classes:

$$y'_c = (1-\epsilon)\cdot y_c + \frac{\epsilon}{K} \tag{10}$$

4.5.4 Adversarial Debiasing

Trains a classifier while preventing encoding of protected attributes via gradient reversal:

$$\mathcal{L} = \mathcal{L}_{cls} - \lambda \mathcal{L}_{adv} \tag{11}$$

4.5.5 Data Scrubbing (Fairness through Blindness)

Directly motivated by our Integrated Gradients analysis, we mask the sensitive tokens identified in Section 4.4. Using the attribution heatmaps (Figures 1 and 2), we construct a set $\mathcal{S}$ of tokens with high attribution magnitude on sensitive attributes:

$$\tilde{x}_i = \begin{cases} \texttt{[MASK]} & \text{if } t_i \in \mathcal{S} \\ x_i & \text{otherwise} \end{cases} \tag{12}$$

The set $\mathcal{S}$ includes city names (Moscow, Saint Petersburg, etc.) and age-related terms (pensioner, student, age numbers).

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Table 1: Comparison of debiasing methods on test set. All methods include inverse square-root reweighting as baseline. Robust metrics use $\tau = 30$. TPR gaps reported as decimals (lower is better for fairness). Best results in bold; worst marked with †.

Model	Acc	F1	Worst Gap	Macro Gap
Baseline (BERT + Sqrt Reweighting)	**0.609**	0.621	0.329	0.116
In-processing methods				
+ Label Smoothing ϵ=0.1	0.596	0.608	0.329	0.140
+ Focal Loss γ=2	0.593	0.613	0.329	0.136
+ Adversarial α=0.5	0.574	0.594	0.429†	0.132
+ GroupDRO η=0.3	0.552	0.572	0.282	0.135
+ GroupDRO η=0.1	0.548	0.553	**0.265**	0.113
Attribution-based methods				
+ Data Scrubbing	0.594	0.611	0.300	**0.112**
+ Attention Reg λ=0.1	0.583	0.601	0.335	0.116

4.5.6 Attention Regularization

We add a penalty term that discourages the model from attending to sensitive tokens:

$$\mathcal{R}_{attr} = \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \sum_{i: t_i \in \mathcal{S}} a_i^{[CLS]} \tag{13}$$

where $a_i^{[CLS]}$ is the average attention weight from the [CLS] token to position i across all layers and heads.

5 Experiments

5.1 Experimental Setup

All models use BERT-base-uncased with learning rate 2×10^{-5}, batch size 8, max sequence length 128, and 2 training epochs.

5.2 Results

Key findings:

GroupDRO achieves the strongest fairness improvement. With $\eta = 0.1$, worst-case TPR gap decreases from 32.9% to 26.5%. However, this comes at a significant accuracy cost (6.1%). The $\eta = 0.3$ variant offers a middle ground with 28.2% worst gap and 5.7% accuracy loss.

Data Scrubbing provides the best accuracy-fairness trade-off. By masking the sensitive tokens identified through our Integrated Gradients analysis (Section 4.4), this method reduces worst-case gap to 30.0% and achieves the best macro gap (11.2%) while losing only 1.5% accuracy. This demonstrates the practical value of interpretability-guided debiasing.

Adversarial debiasing fails catastrophically. Rather than reducing bias, it increases worst-case gap to 42.9%, much worse than the baseline. We hypothesize that with 41 city groups, the adversarial head becomes too difficult to train, leading to unstable gradients.

Focal Loss and Label Smoothing provide minimal fairness benefits while slightly degrading accuracy. These methods primarily address class imbalance rather than group-level disparities.

Attention Regularization shows limited effectiveness. Despite penalizing attention on sensitive tokens, this method slightly worsens fairness (33.5% vs 32.9% baseline), suggesting that attention-level intervention alone is insufficient without representation-level changes.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

6 Discussion

6.1 Why Geographic Bias Dominates

Our multi-attribute analysis revealed that city groups exhibit larger TPR gaps than gender or age. We hypothesize three contributing factors:

1. **Vocabulary variation**: Regional vocabulary patterns, company names, and institutional references create strong textual signals correlated with geography.
2. **Data imbalance**: The 41 city groups create a much more fragmented attribute space than binary gender, amplifying variance.
3. **Historical hiring patterns**: Geographic concentration of certain industries may create legitimate correlations that nevertheless constitute unfair treatment.

6.2 The Value of Interpretability

Our Integrated Gradients analysis (Section 4.4) served two purposes: (1) confirming that the baseline model encodes bias through specific token attributions, and (2) directly informing the Data Scrubbing method by identifying which tokens to mask. This interpretability-guided approach achieved competitive fairness improvements with minimal accuracy loss, demonstrating that understanding *how* a model is biased enables more targeted mitigation.

6.3 Limitations

- Our dataset is limited to Russian IT resumes; generalization requires validation.
- Post-processing calibration methods were not explored.
- Intersectional analysis (e.g., female candidates from small cities) remains for future work.
- All experiments report single-seed point estimates without confidence intervals or significance testing; reported differences between methods may not be statistically significant given the small subgroup sizes in our evaluation.

7 Conclusion

We presented a comprehensive framework for analyzing and mitigating bias in resume classification. Using Integrated Gradients, we first demonstrated that the baseline BERT model encodes systematic bias through city names and age-related vocabulary. Our analysis of multiple sensitive attributes revealed that geographic bias exhibits the strongest measurable disparities.

Our comparative study of six debiasing techniques yields several key insights:

- **GroupDRO** achieves the best fairness improvement but at significant accuracy cost.
- **Data Scrubbing**, guided by interpretability analysis, offers the best practical trade-off, improving fairness while losing only 1.5% accuracy.
- **Adversarial debiasing** fails with many demographic groups (41 cities), highlighting the importance of method selection based on attribute cardinality.
- **Attention regularization** alone is insufficient; representation-level or input-level changes are more effective.

Future work should explore intersectional analysis, post-processing calibration, and deployment in real-world hiring pipelines.

Acknowledgments

This work was supported by the Academy of Sciences of the Republic of Tatarstan, Contract No. 254/2024-PD

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

REFERENCES

Simon Caton and Christian Haas. Fairness in machine learning: A survey. *ACM Computing Surveys*, 56(7), 2024.

Zhisheng Chen. Ethics and discrimination in artificial intelligence-enabled recruitment practices. *Humanities and Social Sciences Communications*, 10, 2023.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017.

Ketki V Deshpande, Shimei Pan, and James R Foulds. Mitigating demographic bias in ai-based resume filtering. In *Adjunct Publication of the 28th ACM Conference on User Modeling, Adaptation and Personalization*, pp. 268–275. ACM, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S Zemel. Fairness through awareness. *arXiv preprint arXiv:1104.3913*, 2012.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. *arXiv preprint arXiv:1610.02413*, 2016.

Tatsunori B Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. *arXiv preprint arXiv:1806.08010*, 2018.

Max Hort, Zhenpeng Chen, Jie M Zhang, Mark Harman, and Federica Sarro. Bias mitigation for machine learning classifiers: A comprehensive survey. *arXiv preprint arXiv:2207.07068*, 2023.

Alina Köchling and Marius Claus Wehner. Discriminated by an algorithm: A systematic review of discrimination and fairness by algorithmic decision-making in the context of hr recruitment and hr development. *Business Research*, 13:795–848, 2020.

Manish Raghavan, Solon Barocas, Jon Kleinberg, and Karen Levy. Mitigating bias in algorithmic hiring: Evaluating claims and practices. *arXiv preprint arXiv:1906.09208*, 2019.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. *CoRR*, abs/1801.07593, 2018. URL `http://arxiv.org/abs/1801.07593`.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

A Additional Attribution Examples

This appendix presents additional IG analysis results supporting the findings in Section 4.4.

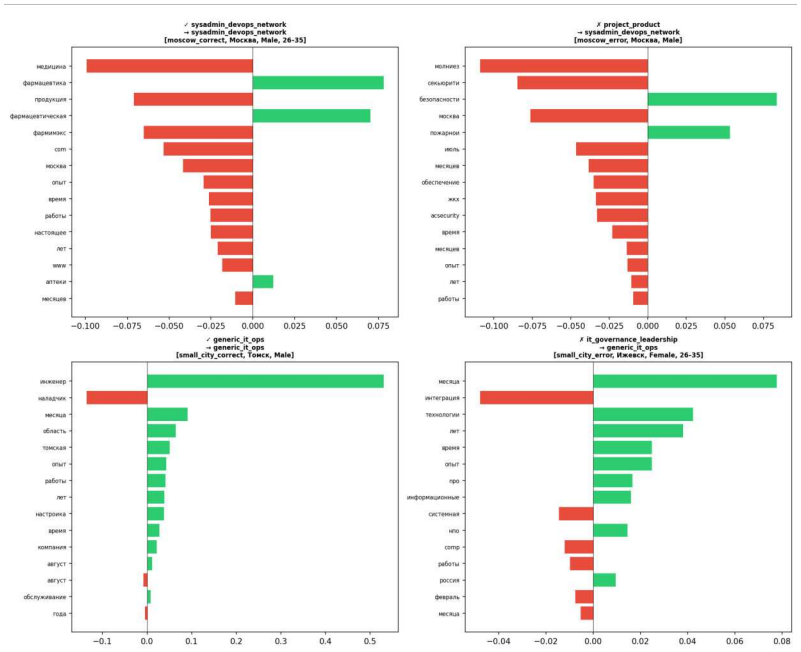


Figure 3: Token attributions: Large vs. small cities.

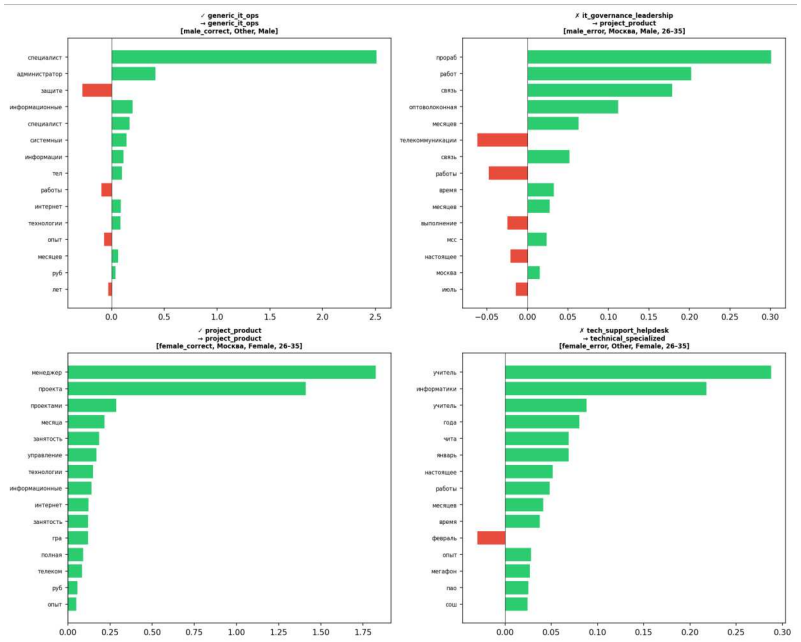


Figure 4: Token attributions by gender.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

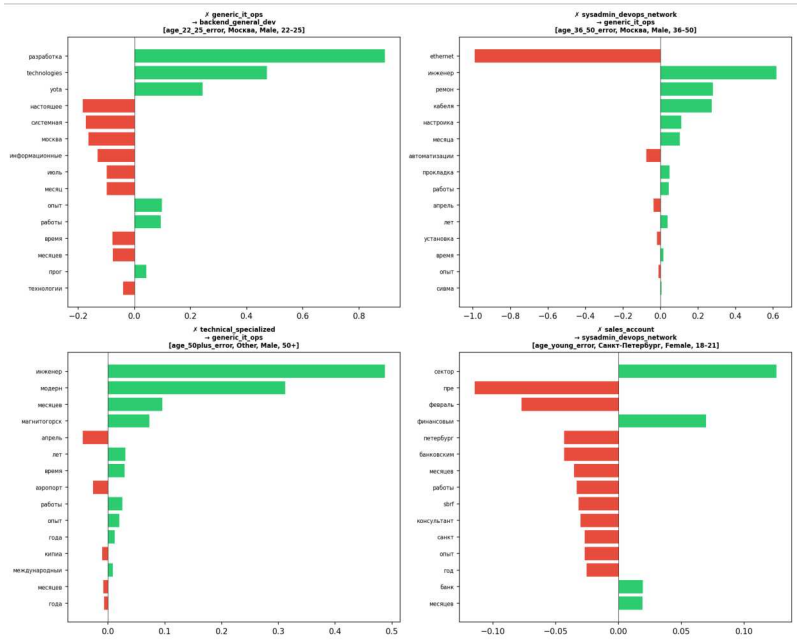


Figure 5: Token attributions across age groups.

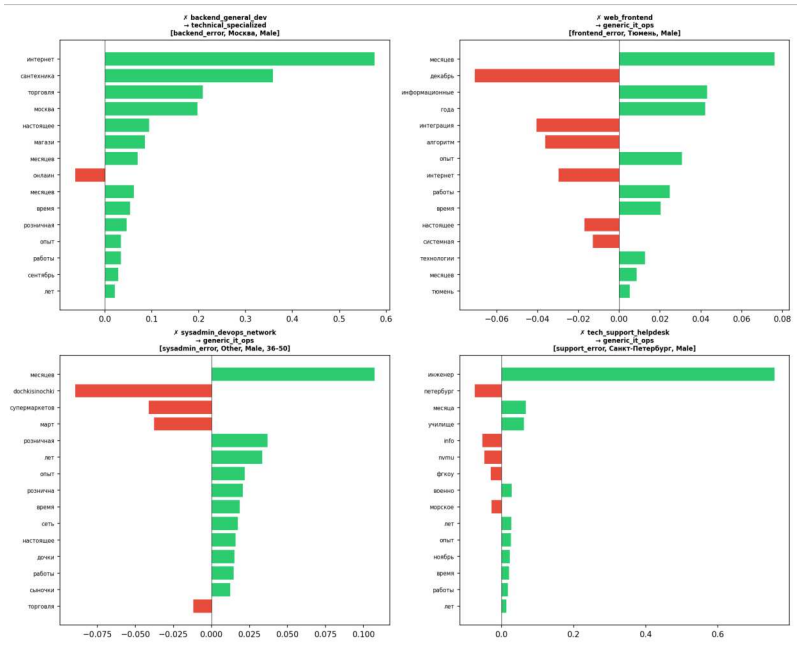


Figure 6: Attribution analysis of profession misclassifications.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1017 DOI: 10.66693/mathai.1017
Special Issue: MathAI 2026 Selected Papers
Received: 20 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

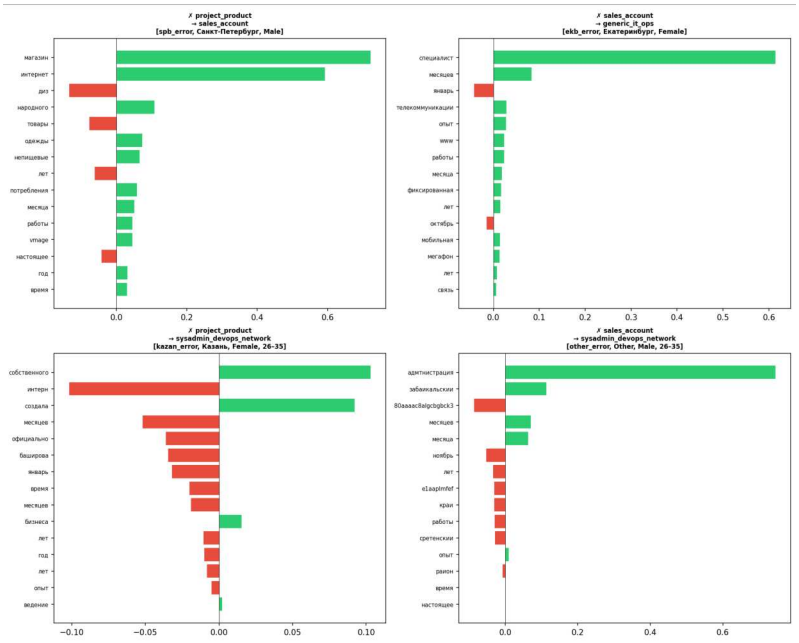


Figure 7: Attribution analysis for regional cities.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

EXPLAINABLE AI FOR MATHEMATICS: PROOFS AS CODE WITH KNOWLEDGE GRAPH AND DOMAIN ONTOLOGY SUPPORT

A. P. Khalov[1,2,*], O. M. Ataeva[1] and N. P. Tuchkova[1]

[1] *Federal Research Center "Computer Science and Control", Russian Academy of Sciences, Moscow, 119333 Russia*
[2] *Moscow Institute of Physics and Technology (National Research University), Dolgoprudny, Moscow Region, 141701 Russia*

khalov.a@phystech.edu, oataeva@frccsc.ru, ntuchkova@frccsc.ru

ABSTRACT

We investigate whether structured knowledge retrieval from a mathematical library's dependency graph can improve neural theorem proving at inference time while maintaining explainability of the retrieved context. Through a controlled ablation study on the miniF2F-Test benchmark (197 tasks, 8 non-chain-of-thought modes, $K{=}8$ attempts per mode), we find that graph-based retrieval augmentation significantly improves proof generation on hard problems – those where the base model's parametric knowledge is insufficient – while having no measurable effect on problems the model already solves reliably. On 109 hard tasks, the best graph-augmented mode nearly triples the baseline success rate (pass@1: $3.56\% \to 10.42\%$, $+6.8$ percentage points, $p{=}0.001$, Wilcoxon signed-rank test). Deterministic pattern-based graph entry points outperform LLM-generated ones ($11\times$ faster, higher accuracy). The retrieved graph context is fully traceable: each hint maps to a specific edge in the Mathlib dependency graph, enabling the user to verify *why* particular lemmas were suggested. The approach is training-free, composable with any base prover, and adds negligible computational overhead.

Keywords: explainable AI, neural theorem proving, knowledge graph, domain ontology, mathlib dependency graph, Lean 4, graph RAG

1 INTRODUCTION

The formalization of mathematical knowledge in interactive theorem provers such as Lean 4 (de Moura and Ullrich, 2021) has created a unique opportunity for AI-assisted mathematics. Under the Curry–Howard correspondence (Howard, 1980; Wadler, 2015), proofs are programs and propositions are types: verifying a mathematical proof reduces to type-checking a program. This equivalence enables rigorous, machine-checkable verification of mathematical reasoning – a property absent from natural-language proofs. In Martin-Löf's intuitionistic type theory (Martin-Löf, 1984), the correspondence extends to dependent types, enabling propositions that quantify over values, and Lean 4 implements precisely this framework. Its mathematical library, Mathlib (The mathlib Community, 2020), contains over 213,000 formally verified declarations, organized as a typed dependency graph where each declaration's type signature and proof body reference other declarations.

Recent neural theorem provers (Yin et al., 2025; Wang et al., 2025; ByteDance Seed AI4Math, 2025) have achieved remarkable pass rates on standard benchmarks by training large language models to generate formal proofs. However, these systems operate as black boxes: when a prover fails, the user receives no insight into *why* the proof attempt was unsuccessful, and when it succeeds, the connection between the generated proof and the underlying mathematical knowledge remains

[*] Corresponding author.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

opaque. This opacity undermines the very premise of formal mathematics – that proofs should be not merely correct, but *understandable*.

We address this problem through *explainable context delivery*: augmenting the prover's prompt with structured hints retrieved from a knowledge graph (KG) built from Mathlib's dependency structure. Each hint is traceable to a specific graph edge – a `usesInType` or `usesInValue` relation between Mathlib declarations – enabling the user to inspect and verify the retrieved context. This approach operates in the *pass@1–3 regime* (Yang et al., 2025): practical settings where computational budgets limit the number of proof attempts, and each attempt should be informed by the best available context. Unlike tree-search provers that require hundreds or thousands of proof attempts, our method provides targeted, interpretable assistance in low-budget scenarios.

Our contributions are as follows. First, we present a **controlled ablation study** comparing 8 non-chain-of-thought modes on miniF2F-Test (Zheng et al., 2022) (197 tasks, $K{=}8$), with paired statistical tests (Wilcoxon signed-rank) and confidence intervals – more rigorous than typical proof generation evaluations. Second, we demonstrate a **strongly asymmetric effect**: graph-augmented context nearly triples the baseline success rate on hard tasks ($+6.8$pp, $p{<}0.001$) while showing no effect on easy tasks, with the hard task set naturally capturing 84–91% of competition-level problems (IMO, AIME). Third, we show that **deterministic pattern-based** graph entry points outperform LLM-generated seeds ($+5.3$pp vs $+2.5$pp, $11\times$ faster, zero LLM calls), demonstrating that simple interpretable rules beat neural generation for knowledge graph navigation. Fourth, our approach is **training-free and composable**, requiring no fine-tuning and applicable on top of any base prover as an inference-time enhancement.

The paper is organized as follows. Section 2 reviews proofs-as-code, neural theorem proving, and retrieval-augmented approaches. Section 3 describes the knowledge graph construction and multimodal representation. Section 4 details the eight context delivery modes. Sections 5–6 present the experimental setup and quantitative results. Section 7 provides a detailed case study demonstrating the explainability mechanism. We discuss implications in Section 8, limitations in Section 9, and conclude in Section 10.

2 Background and Related Work

2.1 Proofs as Code

The Curry–Howard correspondence (Howard, 1980) establishes a deep structural analogy between formal proofs and typed programs: propositions correspond to types, and proofs to terms inhabiting those types. Wadler (Wadler, 2015) provides a modern exposition of this correspondence and its implications for programming language theory and formal verification. Lean 4 (de Moura and Ullrich, 2021) is a dependently typed programming language and interactive theorem prover that implements this correspondence. Its mathematical library, Mathlib (The mathlib Community, 2020), has grown into the largest unified collection of formally verified mathematics, covering algebra, analysis, topology, number theory, and combinatorics. This "proofs as code" perspective has a practical consequence: a proof assistant can *verify* any generated proof by type-checking, regardless of the proof's origin. The verification is sound, complete for the underlying logic, and independent of the generation method, making neural theorem proving uniquely amenable to rigorous evaluation – unlike natural-language mathematics, there is no ambiguity about correctness.

2.2 Neural Theorem Proving

Modern neural theorem provers fall into two broad categories. *Tree-search provers* such as Aristotle (The Harmonic Team, 2025) and Seed-Prover (ByteDance Seed AI4Math, 2025) decompose proofs into tactic steps and explore the proof tree via Monte Carlo tree search or best-first search. *Whole-proof provers* such as DeepSeek-Prover-V2 (Yin et al., 2025) and Kimina-Prover (Wang et al., 2025) generate complete proofs in a single pass, relying on the model's parametric knowledge to produce a valid tactic sequence. Recent advances have been rapid: Goedel-Prover (Lin et al., 2025) achieves 57.6% on miniF2F-Test at pass@32 through synthetic data augmentation and supervised fine-tuning, while DeepSeek-Prover-V2-7B (Yin et al., 2025), building on DeepSeekMath (Shao et al., 2024), reaches 55.5% at pass@1 through reinforcement learning for subgoal decomposition. At higher sampling budgets, Leanabell-Prover-V2 (Wu et al., 2025) achieves 78.2% at pass@128,

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

and Hilbert (Varambally et al., 2025) demonstrates that informal reasoning can guide formal proof construction. Our work focuses on the whole-proof setting, which is more interpretable: the model's output is a single, complete proof that can be read and understood as a coherent argument.

2.3 Retrieval-Augmented Proof Generation

Retrieval-augmented generation (RAG) (Lewis et al., 2020) has proven effective across many knowledge-intensive tasks, and several recent works apply it to theorem proving. REAL-Prover (Shen et al., 2025) fine-tunes a general-purpose model on retrieval-augmented training data, achieving 54.1% on miniF2F-Test at pass@64. LeanDojo (Hsiang et al., 2025) provides infrastructure for premise retrieval in Lean 4 environments. Herald (Gao et al., 2025) constructs a natural-language-annotated Lean 4 dataset for training retrieval models. LeanRAG (Zhang et al., 2025) proposes hierarchical retrieval over knowledge graphs for proof generation. Semantic search engines for Mathlib (Gao et al., 2024; Asher, 2025) enable similarity-based retrieval of relevant declarations, while ATLAS (Liu et al., 2025b) addresses autoformalization as a complementary direction. Our approach differs in three respects: we augment an *already specialized* prover at inference time, requiring no additional training; we retrieve from a *typed dependency graph* rather than a flat document store, providing structured context classified by usage pattern; and we evaluate with *paired statistical tests* on a controlled ablation, isolating the contribution of each retrieval component.

2.4 Knowledge Graphs for Mathematics

Structured representations of mathematical knowledge have a long history, from standards such as MathML (World Wide Web Consortium, 2010) and OpenMath (Buswell et al., 2004) to modern ontological systems. OntoMathPRO (Nevzorova et al., 2014) provides a linked data ontology for mathematical concepts. Serebryakov and Ataeva (Serebryakov and Ataeva, 2021) develop an ontology-based approach to modeling the mathematical domain in digital libraries. The SPAR ontologies (Peroni and Shotton, 2018) and the Open Research Knowledge Graph (Brack et al., 2020) offer frameworks for scholarly knowledge representation more broadly, while formal benchmarks (Zheng et al., 2022; Liu et al., 2025c;a) provide standardized evaluation of mathematical reasoning. AI-assisted scientific research more generally (Chen et al., 2025) has seen rapid progress, though the specific challenge of formal mathematics requires specialized approaches. Our work extends this line by constructing a knowledge graph directly from Mathlib's dependency structure, where nodes are formal declarations and edges represent typed usage relations, capturing the operational structure of mathematical knowledge – which lemmas are used to prove which theorems, and in what capacity.

3 Architecture

3.1 SciLib Ontology and Three-Level Model

The infrastructure is built on the SciLib ontology – a modular meta-level model designed to describe scientific knowledge objects and the processes of their storage, retrieval, and use in computational pipelines. Although in this work only the mathematics domain is considered, the ontology was designed to be domain-general: new subject domains, representations, or computational components can be added without changing the basic semantic invariants. The ontology implements a three-level separation of knowledge structure. The *interpretation level* describes the abstract meaning of a scientific object independently of its form of expression – for instance, the concept "the fundamental theorem of calculus" as understood by a mathematician. The *representation level* fixes the concrete form of expression of this meaning: formal code in Lean 4, natural language text, symbolic notation, or visual rendering. The *resource level* corresponds to the material or computable carrier of representation, including files, database entries, and executable artifacts. Each mathematical statement exists simultaneously at all three levels, linked by a URI that serves as the canonical identifier across modalities. This separation realizes the principle of invariance and enables scalability to an arbitrary number of domains, support for alternative interpretations, and formal analysis of scientific knowledge without mixing levels of abstraction. The three-level model is detailed in a companion submission; here we focus on its role as the foundation for our knowledge graph and multimodal retrieval.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

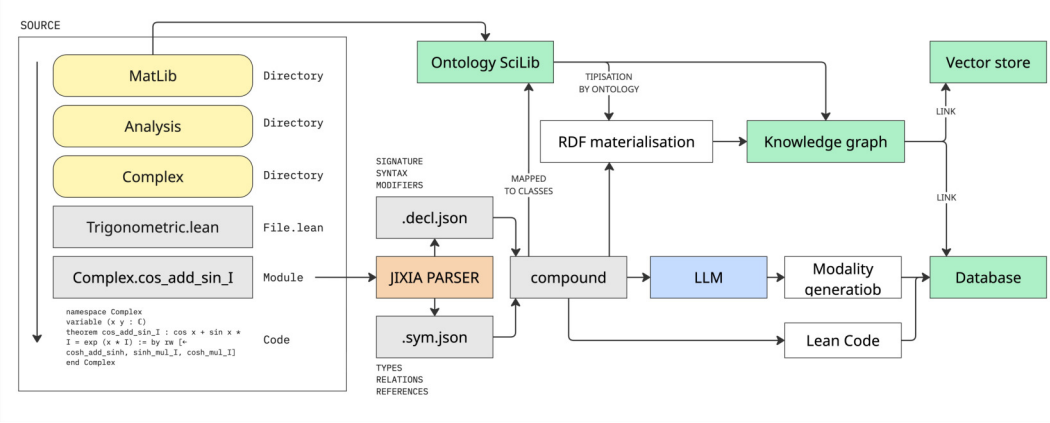


Figure 1: Knowledge graph construction pipeline. Mathlib source is compiled by Lean 4, producing intermediate declaration files. The JIXIA parser extracts declarations, metadata, and structural dependencies, which are mapped to the SciLib ontology and materialized as 66M RDF triples in GraphDB.

3.2 Knowledge Graph Materialization

The knowledge graph is constructed by materializing the Lean 4 Mathlib library into the SciLib ontological model through a reproducible multi-stage pipeline (Figure 1). The Mathlib source is compiled by Lean 4, producing intermediate representations (`.decl.json` and `.sym.json` files). A dedicated parser (JIXIA) extracts three data types from these files: string metadata (names, docstrings), Lean formal code (type signatures, proof bodies), and structural dependencies between declarations. Additionally, the hierarchical organization of the library is analyzed, reflecting the thematic division of mathematical knowledge into 660 subclasses of the Domain class, each automatically annotated using an LLM.

All extracted entities and connections are then mapped to the SciLib ontological model and materialized as an RDF graph. The key methodological assumption is that formal provability in Mathlib serves as a truth criterion: if a statement φ is provable in the formal system, then its interpretation in the mathematical domain is considered true:

$$M_{\text{MathLib}} \vdash \varphi \;\Rightarrow\; I_{\text{Math}}(\varphi) = \text{True} \tag{1}$$

From the structural dependencies, we materialize several types of directed edges. The two most important for retrieval are $\texttt{usesInType}(A, B)$, indicating that declaration A appears in the type signature of B (i.e., A is part of the *statement* of B), and $\texttt{usesInValue}(A, B)$, indicating that A is referenced in the proof body of B (i.e., A is used as a *tactic or lemma* in proving B). Additional relations include `studiedIn` (linking statements to mathematical domains, 180K relations) and `isSpecializationOf` (linking domains hierarchically).

The resulting knowledge graph is substantially larger than the original Mathlib library: starting from approximately 213,000 declarations, the materialization produces over 66 million RDF triples with 6.3 million unique subjects, stored in GraphDB. This $\sim 310\times$ expansion in expressiveness is achieved through ontological typing, reasoner-based inference that supplements the graph with logically derived triples, and the rich structure of the SciLib ontology with its 724 OWL classes. The graph is not a direct copy of Mathlib's internal dependency structure as used, for example, in LeanDojo (Hsiang et al., 2025); it is a substantial extension and reinterpretation of formal mathematical statements through a domain ontology that provides semantic context absent from the raw library.

3.3 Multimodal Embedding Layer

Each Mathlib declaration is represented as a multimodal object linking five modalities through a shared URI (Figure 2): Lean 4 formal code, Russian natural language description, English natural language description, LaTeX symbolic notation, and visual formula rendering. Geometrically, a multimodal object is modeled as a simplex in the embedding space, whose vertices correspond to

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

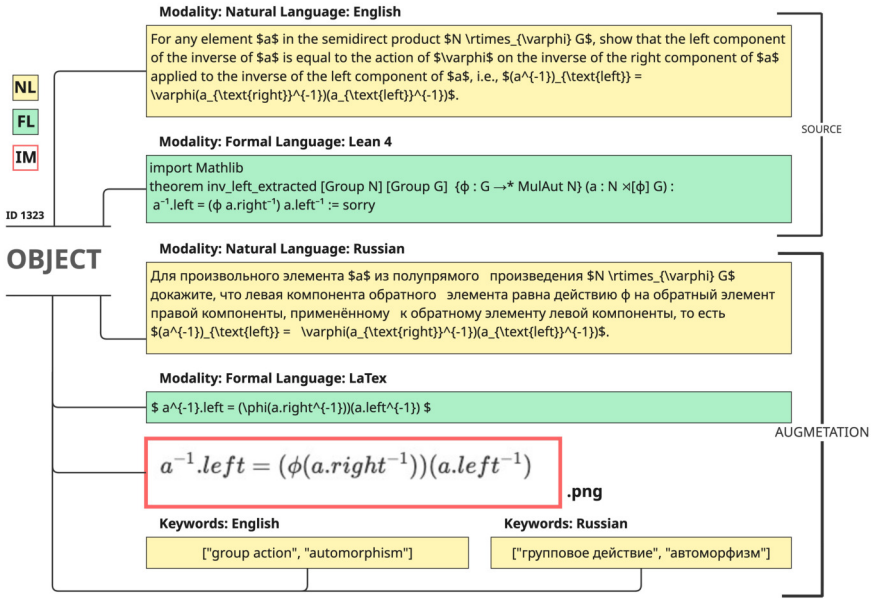


Figure 2: Multimodal representation of a mathematical object. A single Mathlib declaration is linked across five modalities – formal code, Russian and English text, symbolic notation, and visual rendering – through a shared URI, enabling both structured graph traversal and semantic similarity search in a unified vector space.

modalities; the centroid

$$\mu = \frac{1}{M} \sum_{i=1}^{M} z_i \tag{2}$$

serves as an aggregated representation invariant to specific modality (Feng et al., 2014). A dedicated cross-modal encoder, sciLibRuMath, trained on an original dataset of 400,000 mathematical objects, aligns all five modalities into a unified vector space using a combined loss that includes alignment (minimizing simplex volume), contrastive (using centroids as reference vectors with random modality masking), and regularization components. The resulting embedding layer contains over 1 million embeddings across the five modalities, stored in Qdrant, achieving cross-modal Recall@1 above 90% among 10,000 candidates when searching via centroids for most modality pairs. For the experiments in this paper, the vector retrieval mode B1 queries this embedding layer via semantic similarity search, while the graph RAG modes (C11–C23) traverse the RDF knowledge graph directly.

4 Context Delivery Modes

All modes share the same base model (DeepSeek-Prover-V2-7B) and generation parameters. They differ only in how the prompt is constructed before proof generation. We evaluate 8 non-chain-of-thought modes organized in three groups: one baseline, one vector RAG mode, and six graph RAG variants.

4.1 Baseline and Vector RAG

The baseline mode **A0** presents the raw Lean 4 goal to the model with no augmentation – only the theorem statement, necessary imports, and a `sorry` placeholder. This measures the model's parametric knowledge alone. Mode **B1** augments the prompt with lemmas retrieved via semantic search: the model generates candidate Mathlib identifiers (up to 20 LLM calls), each of which

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

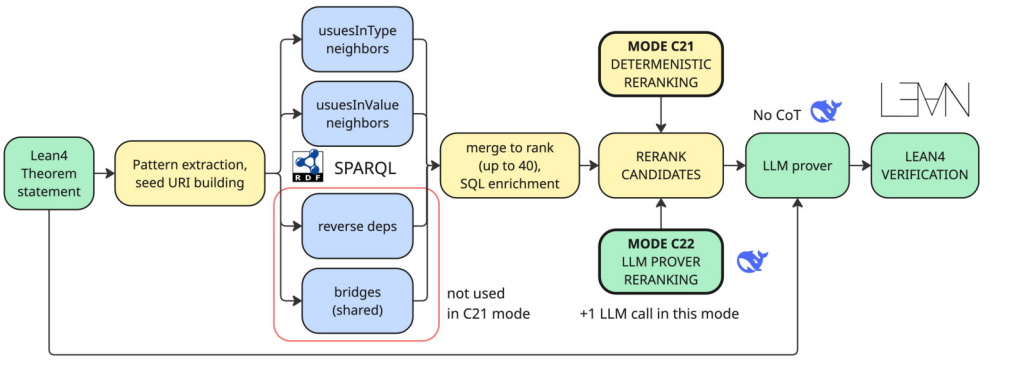


Figure 3: Context delivery pipelines for the two co-leading modes. **C21** (top) combines pattern-based graph expansion with B1 vector retrieval (hybrid). **C23** (bottom) collects candidates from four graph strategies and applies a single LLM reranking call to select the top-8 hints. Both achieve statistically indistinguishable results (p=0.95).

is embedded and queried against the Qdrant collection via the sciLibRuMath encoder. The top-5 nearest neighbors by cosine similarity are appended to the prompt as unstructured hints before the code fence.

4.2 Graph RAG Variants

Six graph RAG modes use the Mathlib dependency graph for context retrieval. They differ along three dimensions: *seed selection* (how graph entry points are chosen), *expansion strategy* (how the graph is traversed), and *vector augmentation* (whether semantic search supplements the graph hints).

Two seed selection approaches are compared. *Model seeds* (modes C1, C2) use an LLM call to generate candidate Mathlib identifiers from the goal text, requiring approximately 30 LLM calls and 130+ seconds per task. *Pattern seeds* (modes C11, C21, C22, C23) use nine regex rules that match goal features – such as inequality operators, `Finset` operations, divisibility predicates, and modular arithmetic symbols – to deterministic Mathlib entry points, requiring zero LLM calls and adding less than one second of overhead. The full list of pattern categories and their mappings is given in Appendix **??**.

From the selected seeds, several expansion strategies are evaluated. **C1** expands model-generated seeds via `usesInType` and `usesInValue` edges. **C11** performs the same typed expansion from pattern seeds. **C22** detects bridge nodes – declarations sharing dependencies with the seed set – identifying structurally related lemmas that may not be direct neighbors. **C23** collects candidates from all four strategies (reverse dependencies, bridges, type neighbors, value neighbors), then applies a single LLM reranking call (64 tokens, greedy decoding) to select the top-8 most relevant hints. Two hybrid modes combine graph and vector retrieval: **C2** merges C1 graph hints with B1 semantic hints (top-3 from each source), and **C21** merges C11 graph hints with B1 semantic hints, leveraging both structural and semantic relevance.

All graph RAG modes format retrieved declarations into four categories reflecting their typical usage in Lean 4 proofs: `apply`/`exact` lemmas applicable as direct proof steps, `rw` lemmas for rewriting equalities, `simp` lemmas for the simplification tactic, and `def` definitions providing vocabulary. This classification makes the retrieved context *actionable*: the model receives not just relevant lemmas, but guidance on *how* to use them. The classification derives from the edge types in the knowledge graph – `usesInType` edges suggest definitional relevance, while `usesInValue` edges suggest proof-body usage. Figure 3 illustrates the pipelines for the two strongest modes.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Table 1: Main results on miniF2F-Test. **All tasks** (197): differences are small and not significant. **Hard tasks** (109, A0 pass@1 $\leq 25\%$): all augmentation modes significantly outperform A0.

	All tasks (197)		Hard tasks (109)			
Mode	pass@1	pass@8	pass@1	Δ A0	p-value	pass@8
A0	41.0	53.8	3.56	–	–	16.5
B1	41.0	55.8	7.34	+3.78	0.0016	20.2
C1	40.9	54.3	6.08	+2.52	0.0135	18.4
C11	42.0	52.0	8.91	+5.32	0.0050	16.7
C2	41.0	54.6	8.10	+4.51	0.0022	19.4
C21	**43.3**	54.1	**10.30**	**+6.71**	**0.0005**	19.4
C22	43.6	52.0	8.68	+5.09	0.0075	15.7
C23	**43.4**	**54.6**	**10.42**	**+6.83**	**0.0011**	**21.3**

5 Experimental Setup

We evaluate on miniF2F-Test (Zheng et al., 2022), a cross-system benchmark of 197 formally stated mathematical problems drawn from AMC, AIME, IMO, and MATH competitions, all formalized in Lean 4. As the base model, we use DeepSeek-Prover-V2-7B (Yin et al., 2025) in non-quantized bf16 precision. Generation parameters are held constant across all modes: temperature T=0.6, top_p=0.95, top_k=40, max_new_tokens=8192, with each task attempted K=8 times per mode.

We define **hard tasks** as those where the unaugmented baseline A0 achieves pass@1 $\leq 25\%$ (at most 2 out of 8 attempts succeed). This yields 109 hard tasks out of 197 total (55%). The threshold is not arbitrary: it identifies tasks where the model's parametric knowledge alone is insufficient. Crucially, it correlates with recognized mathematical difficulty – 91% of AIME problems (10/11), 84% of IMO problems (16/19), and 76% of AMC12 problems (26/34) fall into this category, while only 34% of MATH-D textbook exercises (36/105) do. The threshold thus acts as a principled, model-grounded proxy for mathematical difficulty.

All pairwise mode comparisons use the Wilcoxon signed-rank test (paired, one-sided for directional claims, two-sided for the full matrix in Appendix **??**), computed over per-task pass@1 rates on 109 hard tasks as paired observations. Confidence intervals use the Wilson score method at 95%. Generated proofs are verified by Lean 4 via a dedicated Kafka-based verification service, with each proof compiled against the full Mathlib library using `maxHeartbeats 0`. Results are stored in PostgreSQL with full provenance: attempt ID, mode, timing, raw output, and verification status.

6 Results

6.1 Main Results

Table 1 presents the primary results across all 8 modes, stratified by task difficulty. On all 197 tasks, differences between modes are small (41.0–43.6% pass@1) and not statistically significant. The picture changes dramatically on hard tasks: every augmentation mode significantly outperforms the baseline A0 (all $p < 0.05$), with the best modes nearly tripling the baseline success rate.

6.2 Asymmetric Effect

The central finding is a strongly asymmetric effect (Figure 4). On all tasks, improvements range from $+0.0$ to $+2.1$ percentage points and are not statistically significant ($p > 0.05$). On hard tasks, the same modes yield $+2.5$ to $+6.8$ percentage points, all highly significant ($p < 0.05$). This pattern is consistent with the retrieval-augmentation hypothesis: external knowledge is most valuable when the model's parametric knowledge is insufficient. On easy tasks, the model already has adequate knowledge; additional context is either redundant or occasionally distracting. On hard tasks – predominantly competition-level mathematics – the structured hints from Mathlib's dependency graph provide the missing vocabulary of relevant tactics and lemmas that the model cannot recover from its parameters alone.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

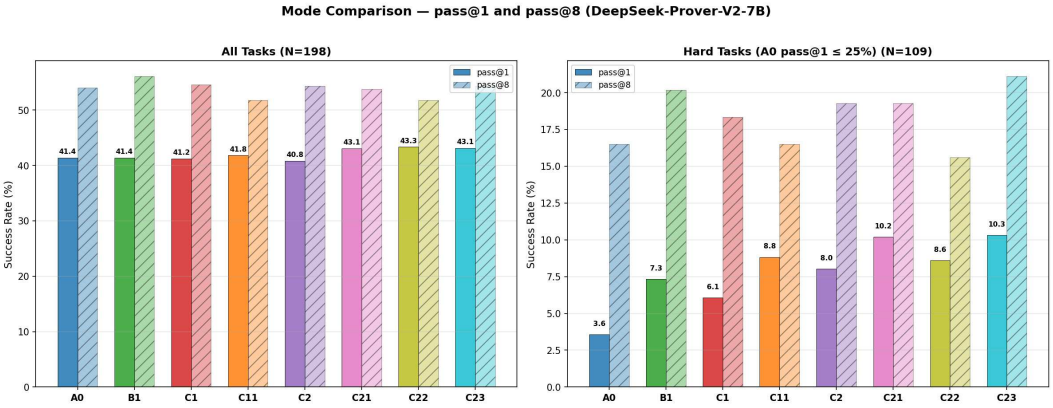


Figure 4: Pass@1 and pass@8 across all modes, stratified by task difficulty. *Left*: on all 197 tasks, modes are indistinguishable. *Right*: on 109 hard tasks, graph RAG modes (C11–C23) substantially outperform the baseline A0. The asymmetry is the key finding.

6.3 Pattern Seeds vs. Model Seeds

Regex-based pattern seeds (C11) achieve pass@1 of 8.91% on hard tasks, significantly outperforming LLM-generated model seeds (C1: 6.08%) with p=0.05. This result is striking given the cost differential: C11 requires zero LLM calls and runs in 12.1 seconds on average, while C1 requires approximately 30 LLM calls and 133.7 seconds ($11\times$ slower). The pattern seeds use nine regex rules matching structural goal features (inequality operators, `Finset` operations, modular arithmetic symbols, and others) to deterministic Mathlib entry points. Their superiority suggests that domain-specific symbolic rules outperform neural text generation for the specific task of selecting entry points into a structured knowledge graph – a finding with implications for the design of retrieval systems in formal mathematics.

6.4 Co-Leaders: C21 and C23

The two strongest modes – C21 and C23 – achieve nearly identical hard-task pass@1 (10.30% vs. 10.42%) through fundamentally different mechanisms. C21 combines pattern-based graph expansion with B1 semantic search, providing both structural and semantic context; the added vector retrieval component accounts for its longer running time (38.4s vs. 12.1s for C11 alone). C23 collects candidates from all four graph strategies, then applies a single LLM reranking call (64 tokens) to select the top-8 hints, trading one cheap LLM call for a more curated context.

The Wilcoxon test between C21 and C23 yields p=0.95 – they are statistically indistinguishable. The convergence of two independent retrieval strategies to the same performance level strengthens the finding: the effect is robust to the specific graph traversal algorithm and likely reflects the intrinsic value of the underlying Mathlib dependency structure. Notably, C21 has stronger statistical significance against A0 (p=0.0005 vs. 0.0011), producing more consistent per-task improvements, while C23 has better pass@8 (21.3% vs. 19.4%) and is $3\times$ faster, producing more diverse correct proofs.

6.5 Coverage Analysis

Beyond per-task improvement, graph RAG expands the set of solvable tasks. Across all 8 modes combined, 118 out of 197 tasks (59.9%) are solved at least once, compared to 106 (53.8%) for A0 alone. Seven tasks are solved exclusively by graph RAG modes (not by A0 or B1), while only two tasks are solved exclusively by the baseline. This asymmetry in exclusive coverage indicates that graph-augmented retrieval unlocks genuinely new problem-solving capabilities rather than merely increasing the probability of solving already-solvable tasks.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

7 Case Study: Product Inequality

We illustrate the mechanism of graph-augmented proof generation through a detailed analysis of a representative hard task: `induction_prod1p1onk3le3m1onn`. This task asks to prove that for all positive integers n:

$$\prod_{k=1}^{n}\left(1+\frac{1}{k^3}\right) \le 3-\frac{1}{n} \tag{3}$$

The Lean 4 formalization is:

```
theorem induction_prod1p1onk3le3m1onn
  (n : ℕ) (h₀ : 0 < n) :
 ∏k ∈Finset.Icc 1 n,
   (1 + (1:ℝ) / k^3) ≤(3:ℝ) - 1 / ↑n := by
 sorry
```

This is a non-trivial inductive inequality involving a finite product over cubic reciprocals. The difficulty lies in the inductive step: splitting the product

$$\prod_{k=1}^{n+1}\left(1+\frac{1}{k^3}\right) = \prod_{k=1}^{n}\left(1+\frac{1}{k^3}\right)\cdot\left(1+\frac{1}{(n+1)^3}\right) \tag{4}$$

and proving that the resulting multiplicative inequality closes correctly with field arithmetic. The contrast between modes is stark: A0 fails all 8 attempts (0/8), while graph-augmented modes succeed up to 75% of the time (C22: 6/8, C21: 5/8, C11: 5/8).

The C21 pipeline extracted pattern seeds from the goal – matching finite product, inequality, and division-by-powers features – and expanded them via `usesInType` and `usesInValue` edges. The structured hints include rewrite lemmas such as `div_le_iff₀` and `le_div_iff₀`, which signal multiplicative inequality manipulation patterns; apply lemmas such as `sq_nonneg` and `le_of_eq`, supporting positivity and equality-based steps; and simp set entries such as `pow_nonneg` and `div_self`, directly used by `field_simp` and `positivity` tactics. The complete hint set is given in Appendix **??**.

The best C21 proof (attempt #57192, verified by Lean 4) uses strong induction, splits the product via `Finset.prod_Icc_succ_top`, and applies `mul_le_mul_of_nonneg_right` to leverage the inductive hypothesis:

```
-- C21 proof (verified):
induction' h₀ with n h₀ ih
-- Base case n=1:
· norm_num
-- Inductive step:
· cases n with
  | zero => contradiction
  | succ n =>
    simp_all [Finset.prod_Icc_succ_top]
    -- KEY: use IH on product prefix
    refine' le_trans
      (mul_le_mul_of_nonneg_right
        ih (by positivity)) _
    cases n with
    | zero => norm_num
    | succ n =>
      field_simp
      refine' le_of_sub_nonneg _
      field_simp; ring_nf; positivity
```

The tactic `mul_le_mul_of_nonneg_right` directly corresponds to the retrieved hints `div_le_iff₀` and `le_div_iff₀`, which signal the multiplicative inequality manipulation pattern. The `field_simp; positivity` closure uses `pow_nonneg` and `div_self` from the simp hint set.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Without graph hints, all 8 A0 attempts adopt a fundamentally flawed strategy: bounding each factor $(1 + 1/k^3)$ individually by 2, yielding $\prod \leq 2^n$, then attempting to prove $2^n \leq 3 - 1/n$ – which is false for $n \geq 2$. The model lacks the vocabulary to recognize that the proof requires strong induction on the product structure and multiplicative inequality manipulation, not per-factor bounding. This example demonstrates the core value proposition: each hint traces to a specific graph edge, and a user inspecting the failed A0 proof alongside the successful C21 proof can identify *exactly which graph-retrieved lemmas* changed the model's strategy.

8 Discussion

The two strongest modes (C21, C23) add minimal overhead to the proof generation pipeline. C23 requires a single LLM call of 64 tokens with greedy decoding and runs in 12.4 seconds on average – comparable to the unaugmented baseline (11.2s). C21, as a hybrid of graph and vector retrieval, runs in 38.4 seconds (comparable to B1 alone at 38.6s), remaining practical for interactive use. By contrast, model-seed modes (C1, C2) require approximately 30–50 LLM calls and 130–160 seconds, yet achieve *lower* hard-task accuracy, suggesting that inference-time cost and retrieval quality are not positively correlated.

Vector RAG (B1) and graph RAG (C11, C22, C23) solve partially disjoint task sets. The two retrieval paradigms operate on different representations – embeddings for semantic similarity, typed edges for structural dependency – and the 7 graph-exclusive versus 2 baseline-exclusive tasks suggest genuine complementarity. The hybrid modes C2 and C21, which combine both sources, confirm this complementarity: C21 achieves the highest statistical significance against A0 (p=0.0005).

The knowledge graph does not "solve" problems – the model does. Rather, the graph provides a *vocabulary of relevant tactics and lemmas* that the model lacks when relying on parametric knowledge alone. This framing is key to understanding both the effect size and the explainability claim: the user can inspect the retrieved vocabulary and understand why the model changed its strategy. The convergence of C21 and C23 (p=0.95) despite their different mechanisms further suggests that the benefit comes from the graph structure itself, not from any specific retrieval algorithm. This is encouraging for practitioners: the choice of expansion strategy matters less than the decision to use the graph at all.

9 Limitations

Our study has several limitations. The sample size of K=8 attempts per task limits per-task statistical power, though results on hard tasks remain robust with 109 paired observations yielding $p < 0.05$ for all modes against A0. Our results are specific to a single model, DeepSeek-Prover-V2-7B; the training-free nature of the approach makes replication with other models straightforward but currently untested. The nine pattern seed categories cover common mathematical structures (inequalities, divisibility, finite sums), but tasks with unusual structures such as combinatorics or geometry may not benefit, and extending pattern coverage is a clear improvement path. The approach requires a typed dependency graph, currently constructed from Lean 4 and Mathlib only; generalization to other proof assistants (Coq, Isabelle) would require analogous graph construction. Finally, we exclude modes with Lean error feedback (R-modes); combining graph RAG with retry loops is a promising but untested configuration.

10 Conclusion

We have shown that structured knowledge retrieval from Mathlib's dependency graph significantly improves neural theorem proving on hard tasks – those where the model's parametric knowledge is insufficient. On 109 hard tasks from miniF2F-Test, the best graph-augmented mode (C23) nearly triples the baseline success rate (pass@1: $3.56\% \rightarrow 10.42\%$, $+6.83$ pp, p=0.001), while adding negligible computational overhead (one LLM call, 12.4 seconds).

Three findings stand out. The effect is strongly asymmetric: graph augmentation helps substantially on hard tasks ($+6.8$ pp, $p<0.001$) while remaining neutral on easy tasks, constraining where retrieval augmentation is useful. Pattern seeds outperform model seeds: deterministic regex-based

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

entry points are $11\times$ faster and more accurate than LLM-generated ones, demonstrating that simple interpretable rules beat neural generation for knowledge graph navigation. The approach is training-free and composable: no model modification is required, and applying graph-augmented context delivery on top of stronger base models (Lin et al., 2025; Wu et al., 2025) is a natural next step.

The key to explainability is traceability: every hint in the augmented prompt maps to a specific edge in the Mathlib dependency graph. The user can inspect which lemmas were retrieved, through which graph path, and how they influenced the generated proof. This transforms neural theorem proving from a black box into a system where human mathematicians can verify not just the proof, but the reasoning behind its construction.

REFERENCES

J. Asher, Asher, J. (2025). LeanExplore: A search engine for Lean 4 declarations. *arXiv preprint arXiv:2506.11085*.

A. Brack, A. Hoppe, M. Stocker, S. Auer, and R. Ewerth, Brack, A. and Hoppe, A. and Stocker, M. and Auer, S. and Ewerth, R. (2020). Requirements analysis for an open research knowledge graph. *arXiv preprint arXiv:2005.10334*.

S. Buswell, O. Caprotti, D. Carlisle, M. Dewar, M. Gaëtano, and M. Kohlhase, Buswell, S. and Caprotti, O. and Carlisle, D. and Dewar, M. and Gaëtano, M. and Kohlhase, M. (2004). The OpenMath standard (version 2.0). Technical report, The OpenMath Society.

ByteDance Seed AI4Math, ByteDance Seed AI4Math (2025). Seed-Prover: Deep and broad reasoning for automated theorem proving. *arXiv preprint arXiv:2507.23726*.

Q. Chen, M. Yang, et al., Chen, Q. and Yang, M. and others (2025). AI4Research: A survey of artificial intelligence for scientific research. *arXiv preprint arXiv:2507.01903*.

Leonardo de Moura and Sebastian Ullrich, de Moura, Leonardo and Ullrich, Sebastian (2021). The Lean 4 theorem prover and programming language. volume and number, Springer.

F. Feng, X. Wang, and R. Li, Feng, F. and Wang, X. and Li, R. (2014). Cross-modal retrieval with correspondence autoencoders. pp., 7–16, ACM.

G. Gao, H. Ju, J. Jiang, Z. Qin, and B. Dong, Gao, G. and Ju, H. and Jiang, J. and Qin, Z. and Dong, B. (2024). A semantic search engine for Mathlib4. *arXiv preprint arXiv:2403.13310*.

G. Gao, Y. Wang, J. Jiang, Q. Gao, Z. Qin, T. Xu, and B. Dong, Gao, G. and Wang, Y. and Jiang, J. and Gao, Q. and Qin, Z. and Xu, T. and Dong, B. (2025). Herald: A natural language annotated Lean 4 dataset. In *International Conference on Learning Representations (ICLR)*.

William A. Howard, Howard, William A. (1980). The formulae-as-types notion of construction. pp., 479–490. Academic Press.

R. Hsiang, W. Adkisson, R. J. George, and A. Anandkumar, Hsiang, R. and Adkisson, W. and George, R. J. and Anandkumar, A. (2025). LeanDojo-v2: A comprehensive library for AI-assisted theorem proving in Lean. In *NeurIPS 2025 Workshop on MATH-AI*.

P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, et al., Lewis, P. and Perez, E. and Piktus, A. and Petroni, F. and Karpukhin, V. and Goyal, N. and Küttler, H. and Lewis, M. and Yih, W. and Rocktäschel, T. and others (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. pp., 9459–9474.

Y. Lin, Z. Chen, H. Wang, Z. Li, J. Song, et al., Lin, Y. and Chen, Z. and Wang, H. and Li, Z. and Song, J. and others (2025). Goedel-Prover: A frontier model for open-source automated theorem proving. *arXiv preprint arXiv:2502.07640*.

Q. Liu, X. Zheng, R. Xia, X. Qi, Q. Cao, and J. Yan, Liu, Q. and Zheng, X. and Xia, R. and Qi, X. and Cao, Q. and Yan, J. (2025a). Beyond theorem proving: Formulation, framework and benchmark for formal problem-solving. *arXiv preprint arXiv:2505.04528*.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020
Special Issue: MathAI 2026 Selected Papers
Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

X. Liu, K. Bao, J. Zhang, Y. Liu, Y. Chen, Y. Liu, Y. Jiao, and T. Luo, Liu, X. and Bao, K. and Zhang, J. and Liu, Y. and Chen, Y. and Liu, Y. and Jiao, Y. and Luo, T. (2025b). ATLAS: Autoformalizing theorems through lifting, augmentation, and synthesis of data. *arXiv preprint arXiv:2502.05567*.

Z. Liu, C. He, K. Zheng, M. Hu, et al., Liu, Z. and He, C. and Zheng, K. and Hu, M. and others (2025c). FormalMATH: Benchmarking formal mathematical reasoning of large language models. *arXiv preprint arXiv:2505.08210*.

(1984). *Intuitionistic Type Theory*. Bibliopolis, Naples.

O. A. Nevzorova, N. Zhiltsov, A. Kirillovich, and E. Lipachev, Nevzorova, O. A. and Zhiltsov, N. and Kirillovich, A. and Lipachev, E. (2014). OntoMathPRO ontology: A linked data hub for mathematics. pp., 105–119. Springer.

S. Peroni and D. Shotton, Peroni, S. and Shotton, D. (2018). The SPAR ontologies. pp., 119–136. Springer.

V. A. Serebryakov and O. M. Ataeva, Serebryakov, V. A. and Ataeva, O. M. (2021). Ontology-based approach to modeling of the subject domain "mathematics" in the digital library. *Lobachevskii Journal of Mathematics*, 42:1920–1934.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. Li, Y. Wu, and D. Guo, Shao, Z. and Wang, P. and Zhu, Q. and Xu, R. and Song, J. and Zhang, M. and Li, Y. and Wu, Y. and Guo, D. (2024). DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Z. Shen, N. Huang, F. Yang, Y. Wang, et al., Shen, Z. and Huang, N. and Yang, F. and Wang, Y. and others (2025). REAL-Prover: Retrieval augmented Lean prover for mathematical reasoning. *arXiv preprint arXiv:2505.20613*.

The Harmonic Team, The Harmonic Team (2025). Aristotle: IMO-level automated theorem proving. *arXiv preprint arXiv:2510.01346*.

The mathlib Community, The mathlib Community (2020). The Lean mathematical library. *arXiv preprint arXiv:1910.09336*.

S. Varambally, T. Voice, Y. Sun, Z. Chen, R. Yu, and K. Ye, Varambally, S. and Voice, T. and Sun, Y. and Chen, Z. and Yu, R. and Ye, K. (2025). Hilbert: Recursively building formal proofs with informal reasoning. In *The 5th Workshop on Mathematical Reasoning and AI (MATH-AI) at NeurIPS 2025*.

Philip Wadler, Wadler, Philip (2015). Propositions as types. *Communications of the ACM*, 58(12):75–84.

H. Wang, Z. Liu, X. Lin, J. Liu, F. Sung, et al., Wang, H. and Liu, Z. and Lin, X. and Liu, J. and Sung, F. and others (2025). Kimina-Prover preview: Towards large formal reasoning models with reinforcement learning. *arXiv preprint arXiv:2504.11354*.

World Wide Web Consortium, World Wide Web Consortium (2010). Mathematical markup language (MathML) version 3.0. Technical report, W3C Working Draft.

X. Wu, H. Wang, Y. Lin, J. Song, et al., Wu, X. and Wang, H. and Lin, Y. and Song, J. and others (2025). Leanabell-Prover-V2: Advancing formal theorem proving via curriculum learning and data augmentation. *arXiv preprint*.

K. Yang, G. Poesia, J. He, W. Li, K. Lauter, S. Chaudhuri, and D. Song, Yang, K. and Poesia, G. and He, J. and Li, W. and Lauter, K. and Chaudhuri, S. and Song, D. (2025). Formal mathematical reasoning – a new frontier in AI. *Position paper, ICML*.

Z. Yin, D. Song, H. Xiong, D. Yu, J. Chen, T. Liu, G. Ma, M. Lin, D. Luo, C. Shao, et al., Yin, Z. and Song, D. and Xiong, H. and Yu, D. and Chen, J. and Liu, T. and Ma, G. and Lin, M. and Luo, D. and Shao, C. and others (2025). DeepSeek-Prover-V2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1020 DOI: 10.66693/mathai.1020

Special Issue: MathAI 2026 Selected Papers

Received: 30 Apr 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Y. Zhang, R. Wu, P. Cai, X. Wang, G. Yan, S. Mao, D. Wang, and B. Shi, Zhang, Y. and Wu, R. and Cai, P. and Wang, X. and Yan, G. and Mao, S. and Wang, D. and Shi, B. (2025). LeanRAG: Knowledge-graph-based generation with semantic aggregation and hierarchical retrieval. *arXiv preprint arXiv:2508.10391*.

K. Zheng, J. M. Han, and S. Polu, Zheng, K. and Han, J. M. and Polu, S. (2022). miniF2F: A cross-system benchmark for formal Olympiad-level mathematics. In *International Conference on Learning Representations (ICLR)*.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1022 DOI: 10.66693/mathai.1022
Special Issue: MathAI 2026 Selected Papers
Received: 04 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

HYBRID UAV HAZARD DETECTION APPROACH BASED ON OPEN-VOCABULARY DETECTION AND MIVAR EXPERT SYSTEM

Oleg Lamcev, Ivan Chernyadiev, Aleksandra Maksimova
Laboratory of Intellegent Systems
Institute of Applied Mathematics and Mechanics
Donetsk, Russian Federation
gelo2003@yandex.ru, {chernyadev-i,maximova.alexandra}@mail.ru

ABSTRACT

The integration of neural network methods in computer vision with logical inference based on a Mivar expert system allows leveraging the advantages of both paradigms: high efficiency in processing unstructured visual data and the interpretability of decisions made based on formalized rules. An analysis of various computer vision tasks was conducted, demonstrating that OVD (Open-Vocabulary Detection) is the preferred tool for dynamic rescue operation scenarios. OVD provides the best balance between the flexibility of detecting arbitrary categories, reliability when working with multiple objects, and the availability of data for training. Modern vision-language architectures, their features, and advantages were investigated. YOLO-World was selected as the base model, as it best meets the stringent requirements of real-time operation, achieving high processing speed while maintaining the flexibility of an open vocabulary. A fine-tuning procedure for the model was carried out, which included freezing the text encoder and the early layers of the convolutional backbone, as well as combining the Flickr30k, VisDrone, and SARD2 datasets. Using Flickr30k helped preserve the quality of the vision-language space, while the specialized datasets adapted the model to real-world application conditions. The fine-tuned model showed a significant increase in accuracy (mAP@50 rose from 0.0974 to 0.342, and mAP@50:95 from 0.0673 to 0.202), and also gained the ability to correctly recognize human poses specific to rescue operations and types of vehicles in drone imagery. A system of parameters and rules for the Mivar Expert System (MES) is proposed, which allows transitioning from simply listing objects in an image to a comprehensive situation assessment. This transforms the system into an active operator assistant, capable not only of detecting but also of interpreting threats. Thus, the developed hybrid intelligent system, combining the YOLO-World detector and the Mivar expert system, fully meets the stated goal and specified requirements:

- Real-time operation due to the optimized architecture;
- Flexibility of control through text prompts in natural language;
- Interpretability and logical validity of decisions thanks to Mivar logical inference.

1 INTRODUCTION

Today's most effective computer vision systems are based on the use of neural network models. The key advantage of such models is their efficient processing of unstructured data and automatic extraction of significant features without manual feature engineering. Traditional machine learning methods are heavily dependent on human data preprocessing. Computer vision algorithms based on neural networks deliver high performance; however, their results are poorly interpretable and prone to hallucinations.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1022 DOI: 10.66693/mathai.1022
Special Issue: MathAI 2026 Selected Papers
Received: 04 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Models built on Multidimensional Informational Variating Adaptive Reality (MIVAR) (Varlamov, 2011) are interpretable. They constitute an expert system that provides logically sound analysis and interpretable decisions. The drawback of the Mivar expert system (MES) is its strictly formalized structure and inability to work with raw data independently. In recent years, a hybrid approach has been gaining popularity. It combines the advantages of both neural network and Mivar methods: feature extraction by the neural network and logical inference using Mivar rules.

Typically, the neural network detector in hybrid systems has a fixed set of classes, making it insufficiently flexible. This poses a problem for dynamic scenarios, such as detecting dangerous situations using unmanned aerial vehicles (UAVs) in rescue operations. The emergence of new relevant objects requires labor-intensive retraining of the model.

This paper proposes a novel hybrid approach that combines a MES with an open-vocabulary object detector. Unlike closed-set models, such a system compares visual and textual features within a unified vector space, expanding capabilities by detecting previously unknown objects using text prompts. This allows the UAV operator to dynamically direct the model's attention to arbitrary objects not included in the training dataset, making this method promising for real-world applications in rapidly changing scenarios.

The aim of this work is to develop a hybrid intelligent system that combines a neural network-based computer vision module and a Mivar expert system. The system is designed for real-time detection of dangerous situations during rescue operations using data obtained from UAVs.

The following key requirements are considered during the system's development:

- Specifics of input data. Images and video streams come from a UAV camera, which determines a top-down or oblique shooting angle, potential dynamic distortions, the influence of lighting and weather conditions, and limited resolution. For fine-tuning the neural network and forming the training set, datasets containing aerial imagery or data captured from drones must be used. This is necessary to ensure adequate generalization capability of the model in real-world operating conditions;
- Real-time operation. The task is dynamic in nature, where the speed of decision-making is critical. The system must process the video stream or sequence of images at a frequency sufficient for prompt response (at least 10-30 frames per second, depending on available computing resources). To meet this requirement, optimized neural network architectures and other inference acceleration methods should be used;
- System control via text prompts in natural language. The UAV operator must be able to specify target object classes or situations to the system through textual descriptions. This functionality allows for the detection of objects and situations not present in the training dataset.

These requirements determine the architectural and algorithmic solutions of the hybrid system, aimed at achieving high efficiency, adaptability, and reliability in real-world rescue operation conditions.

2 Background and related work

The problem is addressed using computer vision methods based on neural networks, combined with a logical approach. Let's consider existing neural network approaches.

Object Detection (OD) is one of the fundamental tasks in computer vision. It involves finding and localizing individual objects within an image or video stream, followed by assigning each detected object a class from a pre-defined, fixed set of categories. The advancement of multimodal approaches has led to the emergence of new object detection paradigms that extend classical detection methods by incorporating natural language descriptions. Let's look at different problem options.

Phrase Grounding (also known as Referring Expression Comprehension or Visual Grounding) is the task of localizing a single object in an image based on a detailed textual description. The model receives an image and a text query that describes one specific object in detail, and must return a bounding box for that object. It is assumed that the described object is definitely present in the image.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1022 DOI: 10.66693/mathai.1022
Special Issue: MathAI 2026 Selected Papers
Received: 04 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Open-Vocabulary Detection (OVD) is an extension of classical object detection where the model is capable of detecting objects from arbitrary categories, specified by short text descriptions in the form of a list of class names. Unlike traditional OD, OVD is not limited to a fixed set of categories defined during the training phase.

Described Object Detection (DOD) (Xie et al., 2023) is a more general and flexible generalization of the REC and OVD tasks. DOD allows the use of language expressions of arbitrary length and complexity: from short category names to detailed descriptions of an object's appearance, context, or attributes. The model must correctly handle cases where the described object is absent from the image, avoiding false positive detections.

2.1 The OVD approach selection

In the formulation of the OVD task, the list of detectable objects is expanded to a list with arbitrary short category names, but it does not account for more complex language descriptions. The REC task, on the other hand, focuses on the precise localization of a single target based on a detailed expression but assumes the target's presence in the image. In real-world scenarios, this assumption is often violated, leading to false positive predictions. The DOD task supports language queries of any complexity, from simple category names to long and detailed descriptions and correctly handles cases where described objects are absent from the image by suppressing false positives.

This work focuses on the OVD task. Compared to REC, which is aimed at localizing a single object based on a unique description, OVD is more effective in scenarios with multiple objects of the same category, supports the absence of objects, does not require unique and potentially ambiguous descriptions, and is closer to classical object detection tasks. Unlike DOD, which supports complex descriptions with attributes and relations, OVD is simpler and more accurate for standard categories, avoids accuracy drops due to long descriptions, and benefits from more mature methods, established benchmarks, and better performance on typical open-vocabulary tasks.

The key advantage of OVD remains the ease of data access: standard object detection datasets (COCO, LVIS, Objects365) are used directly, whereas DOD requires rare and expensive annotations with detailed descriptions, often necessitating complex adaptation or synthesis.

Thus, OVD provides an optimal balance between open-vocabulary flexibility and reliability in detecting multiple objects, requiring significantly lower data preparation costs. It is precisely these advantages that make OVD the preferred approach for dynamic hazard detection systems using UAVs, where it is necessary to promptly detect unauthorized drones, suspicious objects, or multiple threats simultaneously in protected areas, ensuring high response speed, resilience to new types of threats, and minimizing false positives in critical security and monitoring scenarios.

2.2 The role of Vision-Language Models (VLMs) in the OVD task

At the core of OVD lies the use of powerful multimodal models, such as CLIP (Radford et al., 2021) and its analogs. CLIP is trained on vast amounts of image-text pairs using contrastive learning: the model brings the vector embeddings of corresponding pairs closer together and pushes apart those of non-corresponding pairs. This results in the formation of a shared multimodal vector space where visual features are aligned with their textual descriptions for an enhanced visual-semantic representation. This enables the model to generalize to new categories not seen during training, as it can match arbitrary text prompts (class names, descriptive phrases) with visual features.

However, directly applying CLIP to the object detection task encounters a fundamental modality gap. Despite its central role, models like CLIP themselves are not capable of object localization-they only provide a global similarity score. CLIP is trained at the whole-image level, matching it with a global description, whereas detection requires localized understanding. This discrepancy between image-level pre-training and the need for region-level analysis has led to the development of specialized OVD architectures that bridge this gap.

Modern OVD models typically consist of three key components. First, a visual encoder extracts visual features from the input image. While early work relied on convolutional networks (e.g., ResNet), architectures based on transformers, such as the Vision Transformer (ViT) or Swin Trans-

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1022 DOI: 10.66693/mathai.1022
Special Issue: MathAI 2026 Selected Papers
Received: 04 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

former, now dominate. These divide the image into patches, embed them, and process them sequentially, obtaining a dense feature grid where each vector represents a local area.

Second, a text encoder (usually transformer-based, as in CLIP or BERT (Devlin et al., 2019)) encodes the user's text prompt-from simple class names to complex descriptions-into a high-dimensional semantic representation.

Third, a modality fusion mechanism integrates the visual and textual features. Early approaches simply concatenated vectors from different modalities, but modern architectures use more sophisticated methods, particularly cross-attention. In this mechanism, visual features serve as queries for the textual features, allowing the model to dynamically weight and combine information.

The diversity of OVD architectures reflects different strategies for overcoming the modality gap. The choice of a specific implementation determines the balance between accuracy, speed, and generalization capability. Therefore, an analysis of various OVD models will be conducted in the next section.

3 Analysis of existing VLMs for solving OVD

3.1 Florence-2

The Florence-2 model (Xiao et al., 2023), developed by Microsoft researchers, offers a unified approach to solving a wide range of image processing and vision-language interaction tasks based on a single architecture.

The model takes an image and a task-specific text prompt as input, and generates a text response as output, which contains the boundaries of the detected object when necessary. This allows it to solve a broad spectrum of tasks (caption generation, object detection, referring expression comprehension, segmentation, OCR, etc.) using a single architecture and a single set of weights.

The model's architecture includes several key components. The DaViT (Data-efficient Vision Transformer) visual encoder transforms the input image into a sequence of visual representations, effectively capturing spatial and semantic information at different hierarchical levels. The resulting visual representations are projected into a shared representation space and combined with textual representations, which are created from the prompt using an extended language tokenizer and an embedding layer (based on BART). The final multimodal sequence is processed by a transformer encoder-decoder, which generates the output sequence according to the task.

For tasks involving spatial localization, special location tokens representing quantized coordinates have been added to the vocabulary. Supported formats include: rectangle (x0, y0, x1, y1), quadrilateral, and polygon.

Florence-2 was trained in a multi-task learning regime on the FLD-5B dataset (126 million images, 5.4 billion annotations). It embodies the idea of a foundation model for computer vision, similar to trends in natural language processing, where a single architecture and a unified way of representing tasks ensure high knowledge transferability and efficiency across a wide range of applications.

3.2 Kimi-VL

Kimi-VL (Team et al., 2025) is an open vision-language model built on a Mixture-of-Experts (MoE) architecture. It combines high performance in multimodal reasoning tasks with significantly reduced computational costs: with a total of around 16 billion parameters, only about 2.8 billion parameters of the language decoder are activated during inference (in the base Kimi-VL-A3B version).

The model's architecture includes three key components: the MoonViT visual encoder supporting native resolution; an MLP-based projector ensuring efficient alignment of visual and language representations; and an MoE language model that dynamically activates only a small fraction of its parameters. MoonViT, initialized from SigLIP-SO-400M and fine-tuned for high-resolution work, uses an image packing technique and 2D rotary position embeddings, allowing it to process images with resolutions up to several megapixels without the need for fixed-size reshaping.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1022 DOI: 10.66693/mathai.1022
Special Issue: MathAI 2026 Selected Papers
Received: 04 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Kimi-VL demonstrates strong results in complex multimodal tasks: multi-step agent interactions, OCR, image and video understanding, visual mathematical reasoning, and working with multiple images and long videos.

Despite its impressive characteristics, the model has natural limitations due to its scale. In tasks requiring very large volumes of pure linguistic knowledge or solving highly specialized niche problems, its performance lags behind larger models. Furthermore, even with an extended context window (up to 128K tokens), technical challenges related to processing extremely long sequences remain.

3.3 YOLO-World

YOLO-World (Cheng et al., 2024) implements the ”prompt-then-detect” paradigm, which completely eliminates the need for real-time text encoding. User text queries are pre-processed into an offline vocabulary of embeddings, significantly speeding up the inference process and greatly reducing the computational load.

Unlike traditional detectors limited to a fixed set of classes, as well as earlier open-vocabulary methods that use prompt encoding during runtime, YOLO-World combines high speed with flexibility. The detection vocabulary can be dynamically adapted to a specific task without compromising performance, making the model convenient for practical application in real-world scenarios.

The model's architecture includes a detector based on Ultralytics YOLOv8, which extracts multi-scale visual features from the image; a transformer-based text encoder pre-trained within the CLIP framework; and the RepVL-PAN network. The latter is responsible for multi-level cross-modal fusion of visual features and text embeddings. During inference, embeddings from the offline vocabulary are transformed into static weights for the RepVL-PAN network, completely eliminating the need for dynamic text-based computations during runtime.

Three model size variants exist. The small version (YOLO-World-S) has 13 million parameters during inference (77 million in training mode), the medium version (YOLO-World-M) has 29 million parameters (92 million in training), and the large version (YOLO-World-L) has 48 million parameters (110 million in training).

Compared to previous OVD methods, YOLO-World demonstrates a significantly better combination of speed and accuracy. The proposed paradigm and architectural solutions make the model a powerful and practically applicable tool for tasks requiring flexible, fast, and resource-efficient object detection in open-vocabulary scenarios.

4 MIVAR SYSTEM AS LOGICAL PART FOR DECISION MAKING

The Mivar approach represents a modern methodology for developing intelligent systems, based on the use of Mivar knowledge bases and corresponding bipartite network structures. Foundational to this is a unique form of knowledge representation that combines elements of production systems, Petri nets, and entity-relationship models, creating a unified mathematical foundation for logical inference and information processing (Maksimova & Varlamov, 2011).

A key feature of the Mivar approach is its implementation of logical inference with linear computational complexity, even when working with knowledge bases containing a large number of rules and objects. This is achieved through the specialized organization of Mivar networks, in which parameters, rules, and connections are structured as a multidimensional directed graph with clearly separated layers of ”entities” and ”relations.” This approach ensures high scalability, the possibility of efficient parallel processing, and resilience to the increasing complexity of the subject domain.

Mivar systems demonstrate several important advantages compared to traditional expert systems and some modern approaches based on neural networks:

- High adaptability and evolutionary expandability of knowledge;
- Ability to operate under conditions of incompleteness, uncertainty, and inconsistency in initial data;

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1022 DOI: 10.66693/mathai.1022
Special Issue: MathAI 2026 Selected Papers
Received: 04 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

- Support for rapid modification of system behavior without changing the base architecture or retraining models;
- The possibility of explainability of decisions, since inference is built on explicit rules and cause-and-effect relationships.

Mivar technologies have found application in a wide range of subject areas.

In robotics, they are used to build systems for intelligent action planning and autonomous control of complex manipulators and mobile robots. In medicine, Mivar expert systems are applied for differential diagnosis of diseases, forming personalized treatment protocols, and supporting clinical decision-making. In the transportation sector, the Mivar approach enables intelligent control of unmanned vehicles, route optimization, and dynamic planning in changing traffic conditions. Mivar solutions demonstrate significant potential in information security tasks (threat and anomaly analysis), industrial automation (production process management and predictive equipment maintenance), as well as in education-for creating adaptive learning systems, intelligent simulators, and electronic learning environments.

One of the most significant advantages of Mivar expert systems (MES) is their high flexibility when modifying operational logic without the need to redesign the architecture or retrain neural network components. This property is particularly important in dynamically changing subject areas, where requirements for interpreting events or actions may be adjusted during system operation.

Thus, the Mivar approach opens a promising direction for the development of logical artificial intelligence, combining high computational efficiency, explainability of decisions, and unprecedented flexibility in knowledge modification. These characteristics make Mivar technologies especially attractive for creating reliable, scalable, and adaptive intelligent systems in real-world conditions. Further research and practical implementations will allow for a fuller realization of the potential of this approach in solving complex interdisciplinary problems.

5 Adaptation of the neural network to the subject domain

Adaptation of the Neural Network to the Subject Domain. The developed neural network module is based on a pre-trained open-vocabulary model, YOLO-World, in its minimal configuration. This model initially contains 77 million trainable parameters. After removing the text encoder, the text embeddings are transferred to fixed parameters of the classification head, and the number of active parameters during inference is reduced to 13 million. This allows achieving an operating speed of 75 frames per second.

The model accepts input data of two modalities:

- An RGB image obtained from the UAV camera;
- A text query entered by the UAV operator in natural language. These can describe both individual object classes (”person,” ”dog,” ”car”) and their specific attributes and states (”person in a green T-shirt,” ”sitting dog,” ”black SUV”). This formulation enables flexible object detection without needing to retrain the model for each new class or attribute.

To adapt the pre-trained model to the target subject domain (aerial surveying from UAVs, primarily search and rescue tasks, and monitoring of urban/natural environments), a fine-tuning procedure is applied. During this process, the text encoder (based on CLIP) is frozen by default, and additionally, we freeze the first 10 layers of the YOLO model (the backbone network). This reduces the risk of catastrophic forgetting of general vision-language representations and decreases the required computational resources.

A combination of three datasets was used for fine-tuning. Flickr30k is a large dataset of images with detailed text descriptions and bounding boxes. This dataset is used to preserve and maintain the quality of the vision-language space achieved during the pre-training stage. Our own combined dataset, compiled from two open datasets focused on aerial surveying from UAVs: VisDrone and SARD2. This combination of datasets allows training the model to identify various types of vehicles and people in different poses, which is necessary for our subject domain.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1022 DOI: 10.66693/mathai.1022
Special Issue: MathAI 2026 Selected Papers
Received: 04 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

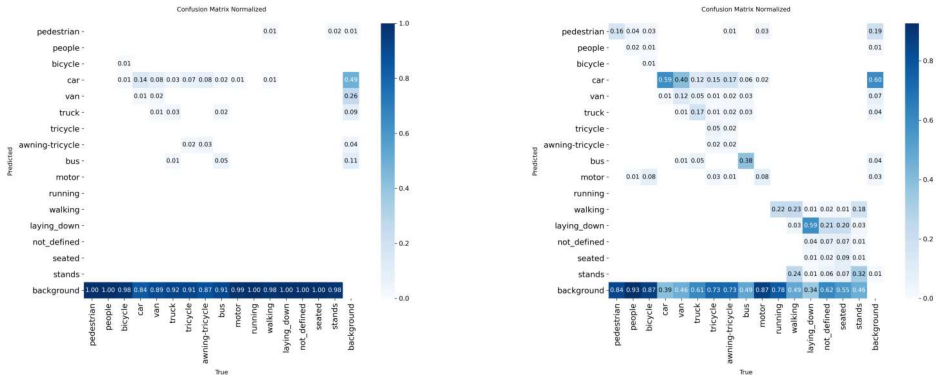
The following approach was used for fine-tuning the model: fine-tuning was performed on our own combined dataset (10,609 images) for the selected subject domain. From these, 7,857 were allocated for training, 944 for validation, and 1,808 for testing. In these datasets, integer class labels are replaced with string class names to bring the data into a format compatible with the open-vocabulary paradigm. The Flickr30k dataset was used to connect textual and visual embeddings in a generalized vector space.

Thus, the combination of Flickr30k and the target datasets ensures a balance between preserving the model's generalization capability and its specialization for objects and scenes typical of unmanned reconnaissance and aerial monitoring tasks.

6 Description of Results and Metrics

To evaluate the effectiveness of the neural network model, test images from the VisDrone and SARD2 datasets were used. Figure 1 presents two normalized confusion matrices: the matrix on the left corresponds to the pre-trained model, and the one on the right corresponds to the model fine-tuned on our custom dataset. The pre-trained model demonstrated metrics of 0.0974 (mAP@50) and 0.0673 (mAP@50:95). After fine-tuning, the model's performance significantly improved, reaching 0.342 and 0.202 on the respective metrics.

As a result of fine-tuning, the model not only increased detection accuracy but also became capable of recognizing people in various poses (walking, lying, sitting, standing) and types of vehicles in images obtained from UAVs. The output of the pre-trained and fine-tuned models is shown in Figure 2 and Figure 3.



(a) (b)

Figure 1: Confusion matrices of the pre-trained model (a) and the fine-tuned model (b).



(a) (b)

Figure 2: Output of the pre-trained model (a) and the fine-tuned model (b) for identifying types of vehicles.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1022 DOI: 10.66693/mathai.1022
Special Issue: MathAI 2026 Selected Papers
Received: 04 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026



(a) (b)

Figure 3: Output of the pre-trained model (a) and the fine-tuned model (b) for determining human poses.

7 Conclusions

The hybrid system, combining the MES with an open-vocabulary object detector, opens a new path for intelligent support of rescue operations using UAVs. In the future, such a paradigm could become the foundation for universal intelligent platforms that automatically adapt to new types of threats and scenarios through dynamic expansion of the vocabulary and rules. This is particularly valuable in conditions of uncertainty, contradictory data, and rapidly changing environments.

Further research may be directed towards significantly expanding the base of Mivar rules to cover natural disasters, technogenic accidents, and complex multifactor emergency situations. Integrating the system with UAV telemetry (flight altitude, GPS coordinates, heading) and camera orientation appears to be a promising direction, as it would enable a comprehensive spatiotemporal analysis of the observed scene. Furthermore, the developed architecture has high potential for adaptation to tasks in information security, industrial monitoring of critical infrastructure, and autonomous robotic systems, where high response speed, flexibility, and complete transparency of decisions made are simultaneously required.

References

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection, 2024. URL https://arxiv.org/abs/2401.17270.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL https://arxiv.org/abs/1810.04805.

Aleksandra Yu. Maksimova and Oleg O. Varlamov. A mivar expert system for pattern recognition based on fuzzy classification and modeling of various subject areas with automated context expansion, 2011.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, Congcong Wang, Dehao Zhang, Dikang Du, Dongliang Wang, Enming Yuan, Enzhe Lu, Fang Li, Flood Sung, Guangda Wei, Guokun Lai, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haoning Wu, Haotian Yao, Haoyu Lu, Heng Wang, Hongcheng Gao, Huabin Zheng, Jiaming Li, Jianlin Su, Jianzhou Wang, Jiaqi Deng, Jiezhong Qiu, Jin Xie, Jinhong Wang, Jingyuan Liu, Junjie Yan, Kun Ouyang, Liang Chen, Lin Sui, Longhui Yu, Mengfan Dong, Mengnan Dong, Nuo Xu, Pengyu Cheng, Qizheng Gu, Runjie Zhou, Shaowei Liu, Sihan Cao, Tao Yu, Tianhui Song, Tongtong Bai, Wei Song, Weiran He, Weixiao Huang, Weixin

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1022 DOI: 10.66693/mathai.1022
Special Issue: MathAI 2026 Selected Papers
Received: 04 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Xu, Xiaokun Yuan, Xingcheng Yao, Xingzhe Wu, Xinhao Li, Xinxing Zu, Xinyu Zhou, Xinyuan Wang, Y. Charles, Yan Zhong, Yang Li, Yangyang Hu, Yanru Chen, Yejie Wang, Yibo Liu, Yibo Miao, Yidao Qin, Yimin Chen, Yiping Bao, Yiqin Wang, Yongsheng Kang, Yuanxin Liu, Yuhao Dong, Yulun Du, Yuxin Wu, Yuzhi Wang, Yuzi Yan, Zaida Zhou, Zhaowei Li, Zhejun Jiang, Zheng Zhang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Zijia Zhao, Ziwei Chen, and Zongyu Lin. Kimi-vl technical report, 2025. URL https://arxiv.org/abs/2504.07491.

Oleg O. Varlamov. Mivar: Transition from productions to bipartite graphs mivar nets and practical realization of automated constructor of algorithms handling more than three million production rules, 2011. URL https://arxiv.org/abs/1111.1321.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks, 2023. URL https://arxiv.org/abs/2311.06242.

Chi Xie, Zhao Zhang, Yixuan Wu, Feng Zhu, Rui Zhao, and Shuang Liang. Described object detection: Liberating object detection with flexible expressions, 2023. URL https://arxiv.org/abs/2307.12813.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Implementation of a Cryptographic Hash Function Based on a Deep Neural Network

Dmitry V. Iatsenko
Southern Federal University
Rostov-on-Don, Russia
d.yacenko@gmail.com

Denis S. Gorin
RTU MIREA
Moscow, Russia
gorin@mirea.ru

Abstract

We present a two-layer construction for image hashing. First, a perceptual binary code $c(x)$ is derived from a ResNet-18 embedding (after global average pooling, $d = 512$) via a linear projection and sign quantization; optionally, a real-valued serialization of length $n = 8ds$ bits is used. The code $c(x)$ enables fast approximate nearest-neighbor search. Second, $c(x)$ serves as a noisy source for a fuzzy extractor producing a reproducible secret R and public data P; a cryptographic tag T is then derived via KDF and HMAC/SHA-3. This preserves similarity search over $c(x)$ while assigning cryptographic guarantees (preimage/second-preimage/collision) to T, which reduce to the security of the underlying primitives given sufficient post-publication min-entropy $H_\infty(C \mid P)$. We also report preliminary proof-of-concept experiments on a small demo dataset: we measure intra-BER under admissible transforms, inter-image Hamming distances, and successful FE/tag verification on retrieved candidates. These experiments are intended to validate the mechanics of the proposed pipeline rather than provide a large-scale benchmark.

Keywords: perceptual hash; ResNet-18; feature binarization; fuzzy extractor; min-entropy; approximate nearest neighbor; HMAC; SHA-3; error-correcting codes; cryptographic tag.

1 Introduction

Image matching systems often require two different properties at once. On the one hand, they must support similarity search: visually similar images should map to nearby codes. On the other hand, they may require a stronger verification mechanism that is stable under admissible distortions but still grounded in standard cryptographic primitives. These two goals are different in nature: perceptual hashing is useful for retrieval, while cryptographic hashing is useful for strict verification.

This paper studies a two-layer construction that separates these roles. First, a deep visual embedding is converted into a binary perceptual code suitable for approximate nearest-neighbor search in Hamming space. Second, this code is treated as a noisy source for a fuzzy extractor, from which we derive a reproducible secret and a cryptographic tag. Thus, the perceptual layer supports search, while the cryptographic layer supports verification.

The main contribution of the paper is a compact implementation-oriented formulation of this pipeline together with a preliminary proof-of-concept evaluation. We explicitly distinguish what is claimed at the perceptual-code level and what is claimed only after the fuzzy-extractor and tag-generation stages. In particular, we do not claim that the perceptual code itself is a cryptographic hash; rather, cryptographic guarantees are assigned only to the final tag.

Terminology and metrics. Throughout the paper, BER denotes the bit error rate between two binary codes. FRR (false reject rate) is the probability that a genuinely matching query fails verification. FAR (false accept rate) is the probability that a non-matching query is

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

accepted. We also briefly refer to the classical cryptographic criteria SAC (strict avalanche criterion) and BIC (bit independence criterion) only for contrast with perceptual hashing; they are not required of the perceptual code used in our construction.

2 Perceptual Hash Layer

2.1 Perceptual hash based on ResNet-18

We consider an embedding based on the ResNet-18 convolutional network with the final fully connected layer removed (He et al., 2016). Let $e(x) \in \mathbb{R}^d$ be the output feature vector after global average pooling (for the standard ResNet-18, $d = 512$ is typical). We then construct from $e(x)$ a perceptual hash intended for searching visually similar images.

Canonical preprocessing. For reproducibility we fix a pipeline: resizing to 256×256, center crop 224×224, channel normalization (μ, σ), and fixed byte order and image format. This minimizes feature drift due to input ambiguities.

Definitions. We use two representations:

1. Real-valued descriptor $v(x) = \mathrm{norm}(e(x)) \in \mathbb{R}^d$ (L_2 normalization; optionally whitening/PCA).
2. Binary perceptual hash $c(x) \in \{0,1\}^n$—a quantized code for fast comparison in Hamming space.

Binary hash construction. Let $A \in \mathbb{R}^{n \times d}$ and $b \in \mathbb{R}^n$ be fixed public projection/centering parameters chosen on a training set. Possible options include random hyperplanes (p-stable LSH (Datar et al., 2004)), learned orthogonal/projection schemes (ITQ (Gong & Lazebnik, 2011)), and quantization from the PQ/OPQ family for compact indices (Jégou et al., 2011; Ge et al., 2013). Define $z(x) = A\,v(x) + b$ and sign binarization

$$c_i(x) = \mathbf{1}\{z_i(x) \geq 0\}, \quad i = 1, \ldots, n.$$

Typically $n \in [256, 2048]$: increasing n improves discriminativeness, but increases the average BER between "related" instances, which is important for the subsequent use of a fuzzy extractor.

Remark on serialization. If needed, the real-valued descriptor $v(x)$ is stored in a fixed format (float16/float32); its length is $8ds$ bits. This is not a cryptographic hash, but an auxiliary representation for search/re-quantization.

2.2 Properties of the perceptual hash

Stability (robustness). A family of admissible transforms $\mathcal{T}$ is specified (moderate geometric/photometric distortions, compression). For $T \in \mathcal{T}$ we evaluate $\mathrm{BER}(x, Tx)$. We require $\mathbb{E}_{x,T}[\mathrm{BER}(x, Tx)] \leq \varepsilon$ (e.g., < 0.1).

Discriminativeness. For pairs of "different" images (x, x'), the distribution of $\mathrm{Ham}(c(x), c(x'))$ should be substantially shifted to the right compared to "same/similar" pairs; we evaluate ROC/mAP for ANN search in Hamming space (Gong & Lazebnik, 2011; Jégou et al., 2011; Ge et al., 2013).

Bit balance and weak inter-bit correlations. We would like $\Pr[c_i = 1] \approx \frac{1}{2}$ and small $|\mathrm{corr}(c_i, c_j)|$ for $i \neq j$. Balance can be achieved by selecting offsets b and/or thresholds on data.

Collisions (perceptual). Collisions are expected and acceptable: code matches should, with high probability, correspond to perceptually similar images. The theoretical estimate $\approx 2^{-n}$ applies only for independent balanced bits; in practice we use empirical frequencies.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Hardware-software stability. To reproduce the binary code we fix the model/library versions and numeric precision; sign binarization after a linear projection reduces sensitivity to floating-point implementation details.

2.3 Limitations and threat model

The perceptual hash $c(x)$ is not a cryptographic hash in the sense of (Dang, 2012; Menezes et al., 1996): it is deterministic, unkeyed, allows targeted collisions, and is vulnerable to adversarial modifications that preserve the code (Struppek et al., 2021). Pre-image/second-pre-image/collision resistance properties are not claimed at this level.

2.4 Preparing to apply a fuzzy extractor

In the second part, we build a fuzzy extractor on top of $c(x)$ (Dodis et al., 2008). Already at this stage we fix the source characteristics required for the FE:

- Bit error for "similar" instances: we estimate the distribution of $\mathrm{BER}(x, Tx)$ and choose n/thresholds so that an error-correcting code capable of correcting t errors exists with acceptable false reject rate (FRR) and false accept rate (FAR).
- Min-entropy: we empirically estimate a lower bound on $H_\infty(C)$ and approximate independence/balance of bits to ensure sufficient entropy of the extracted secret.
- Public data: parameters (A, b) and FE sketch data are public; we then analyze leakage $H_\infty(C \mid P)$.

2.5 Selective sensitivity: robustness within class and "avalanche" between classes

The classical cryptographic criteria—the strict avalanche criterion (SAC) and the bit independence criterion (BIC)—(Webster & Tavares, 1986; Cusick, 1994; Menezes et al., 1996) require that flipping a single input bit changes, on average, about half of the output bits. This objective is relevant for cryptographic hashes, but it does not coincide with the goals of perceptual hashing, where the priority is stability to small perceptual transforms and separability of dissimilar images. Therefore, below we use the term "selective sensitivity" instead of "avalanche effect" and distinguish two regimes:

(i) Robustness within class. We specify a family of admissible transforms $\mathcal{T}$ (moderate geometric/photometric changes, compression, etc.). For $T \in \mathcal{T}$ we evaluate

$$\mathrm{BER}_{\mathrm{intra}}(x, Tx) \;=\; \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{c_i(x) \neq c_i(Tx)\}.$$

We require $\mathbb{E}_{x,T}[\mathrm{BER}_{\mathrm{intra}}] \leq \varepsilon$ for small ε (e.g., < 0.1), ensuring code reproducibility for similar images. This quantity is critical for the subsequent fuzzy extractor, since it determines the required error-correction capability (radius t).

(ii) "Avalanche" between classes. For pairs of dissimilar images (x, x'), the classical SAC/BIC ideal for cryptographic hashes would correspond to an almost random behavior with normalized Hamming distance near $\frac{1}{2}$. Our perceptual code does not aim to satisfy this strict cryptographic criterion literally; rather, we seek an avalanche-like between-class effect: dissimilar images should induce substantially larger Hamming distances than near-duplicate or admissibly transformed pairs, thereby reducing the probability of false matches.

Practical check. We run two experiments:

- Intra robustness curve: for each test x and a set of $T \in \mathcal{T}$, we build a histogram of $\mathrm{BER}_{\mathrm{intra}}$; we report μ, σ and the quantile $q_{0.95}$—this directly sets the required correction radius $t \approx n \cdot q_{0.95}$ for the FE.
- Inter separability curve: for random pairs (x, x'), we build the distribution of $\mathrm{Ham}(c(x), c(x'))/n$; ideally, the median is close to 0.5 and the mass near $[0, 0.3]$ is small.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Bit balance and weak correlation. To increase the entropy of the source $C = c(X)$, we prefer $\Pr[c_i = 1] \approx \frac{1}{2}$ and small $|\mathrm{corr}(c_i, c_j)|$ for $i \neq j$; this is achieved by selecting offsets/thresholds and (if needed) orthogonal projections (Datar et al., 2004; Gong & Lazebnik, 2011). These properties matter not only for search quality, but also for a lower bound on the min-entropy used by the fuzzy extractor (Dodis et al., 2008).

Remark on vulnerabilities. Neural-network codes are vulnerable to targeted adversarial changes that can preserve or imitate the code (Struppek et al., 2021); this is an expected limitation of perceptual hashing and does not contradict the stated goals. Cryptographic security is provided at the next step by the fuzzy extractor and the subsequent use of standard cryptographic primitives (KDF and AEAD). Therefore, at the perceptual-hash level it is sufficient to require robustness to uncontrolled input distortions, rather than cryptographic resistance.

2.6 Perceptual collisions and empirical separability

Within perceptual hashing, by a collision we mean equality of binary codes $c(x) = c(x')$ or their "sticking" within a Hamming radius r, $\mathrm{Ham}(c(x), c(x')) \leq r$, for different and dissimilar images $x \neq x'$. The goal is for such events to be rare, and for the distance distribution for dissimilar pairs to be shifted to the right relative to near-duplicate pairs, with the idealized random-hash regime near $n/2$ serving only as a conceptual reference point.

Metrics.

1. Inter-distribution: the distribution D_{inter} of $\mathrm{Ham}(c(x), c(x'))/n$ for random dissimilar pairs (x, x'). Ideally, the median ≈ 0.5 and the left mass is small.
2. Probability of a random match under threshold r: $p_{\leq r} = \Pr[\mathrm{Ham}(c(x), c(x')) \leq r]$ under D_{inter}. For a collection of size M, the expected number of false matches is $\mathbb{E}[\mathrm{FP}] \approx \binom{M}{2} p_{\leq r}$.
3. Balance and correlations: we empirically estimate $\Pr[c_i = 1]$ and $\mathrm{corr}(c_i, c_j)$. Balance $\approx 1/2$ and weak inter-bit correlations increase the source entropy and reduce the risk of perceptual collisions (Gong & Lazebnik, 2011; Jégou et al., 2011; Ge et al., 2013; Datar et al., 2004).

Why we do not use the "birthday bound" directly. The classical bound $2^{n/2}$ for collision search holds for an ideal random hash into $\{0,1\}^n$ (Menezes et al., 1996). Our code $c(x)$ is constructed from an embedding and a thresholded linear projection; bits can be biased and dependent. Therefore, we rely on empirical D_{inter} and $p_{\leq r}$ rather than an IID model.

Attacks and assumptions. In a white-box setting, targeted construction of collisions for $c(\cdot)$ via optimization (adversarial techniques) is possible, consistent with known vulnerabilities of neural perceptual features (Struppek et al., 2021). This does not contradict the purpose of this section: here we evaluate perceptual properties and the probability of random matches. Cryptographic collision resistance is provided at the next layer (see below).

Relation to the fuzzy extractor. To build an FE we need: (i) low intra-BER for "own" pairs (see the previous section) and (ii) sufficient min-entropy of the source $C = c(X)$. Min-entropy is estimated from code/cluster frequencies and bit balance; then the extracted secret length R is constrained by standard relations (residual randomness after publishing helper data), see (Dodis et al., 2008). Collisions $c(x) = c(x')$ for dissimilar x, x' are acceptable to the extent that they are rare and do not reduce the min-entropy below the threshold required for the target key length.

3 Cryptographic tag over a perceptual hash via a fuzzy extractor

In the previous sections we constructed a binary perceptual code $c(x) \in \{0,1\}^n$ with low intra-BER for "own" transformations and an avalanche-like right-shifted inter-distance dis-

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

tribution for "others". We now use it as a "noisy source" in a fuzzy extractor (FE) (Dodis et al., 2008) to obtain a reproducible secret R and public helper data P, and then produce a cryptographic tag T.

3.1 Construction scheme

Components.

- Perceptual code: $C = c(X) \in \{0,1\}^n$ (see above).
- FE algorithms: Gen : $\{0,1\}^n \to \{0,1\}^\ell \times \mathcal{P}$ and Rep : $\{0,1\}^n \times \mathcal{P} \to \{0,1\}^\ell$ such that, if $\mathrm{dist}_H(C, C') \le t$, then with high probability $\mathrm{Rep}(C', P) = R$ where $(R, P) = \mathrm{Gen}(C)$ (Dodis et al., 2008).
- Extractor/KDF: KDF (e.g., HKDF over SHA-3).
- Tag primitive: HMAC or SHA-3 (depending on the application) (Menezes et al., 1996).

Enrollment.

1. Compute $c = c(x)$.
2. $(R, P) \leftarrow \mathrm{Gen}(c)$.
3. Compute the key $K \leftarrow \mathrm{KDF}(R \,\|\, \mathrm{ctx})$ with domain separation string ctx.
4. Form the record tag $T \leftarrow \text{SHA3-256}(K \,\|\, \mathrm{meta})$ or $T \leftarrow \mathrm{HMAC}_K(\mathrm{meta})$.

Store in the database: the index code for search c or its compact/LSH representation, the public data P, and the tag T. (Under stricter privacy requirements one may store only LSH signatures instead of the full c.)

Query (search/verification).

1. For a query x', compute $c' = c(x')$.
2. Similarity search: use the index (Hamming/ANN) to find candidates with $\mathrm{dist}_H(c', c) \le r$.
3. Candidate verification: for each returned record, reconstruct $R' = \mathrm{Rep}(c', P)$, then compute $K' = \mathrm{KDF}(R' \,\|\, \mathrm{ctx})$ and T', and compare T' with the stored T; equality confirms "identity by tag".

3.2 Properties and guarantees

Compatibility with search. Indexing and nearest-neighbor selection are performed using the perceptual code c (or LSH), i.e., the ability to find similar images is preserved. The FE/cryptographic part is used only at the candidate verification stage and does not "break" the metric.

Cryptographic security of the tag. The tag T is built from $K = \mathrm{KDF}(R, \mathrm{ctx})$ where R is the FE output. Under sufficient source min-entropy after publishing P (denote $\tilde{H}_\infty = H_\infty(C \,|\, P)$), the length ℓ is chosen with a margin according to standard FE estimates (see (Dodis et al., 2008)). Then the security of T (preimage/second-preimage/collision) reduces to the security of the underlying primitive (SHA-3/HMAC) and the unpredictability of R:

- Collision resistance: finding two inputs that produce the same T is no easier than breaking the collision resistance of the chosen primitive (for fixed ctx/meta).
- Preimage/2nd-preimage: recovering R from T or finding another R' with the same T reduces to breaking the primitive; obtaining a corresponding c' additionally requires meeting the FE tolerance (otherwise Rep will not reconstruct the original R).

Important: "tag equality" does not mean bitwise equality of the original pixels; the FE intentionally allows variations controlled by the threshold t.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Parameters and code selection.

- We choose the length n and binarization thresholds from measured BER distributions so that the 95th percentile of intra-BER fits into the admissible radius t/n for the selected error-correcting code (BCH/LDPC/RS).
- We choose the secret length $\ell \leq \tilde{H}_\infty - \text{losses} - \text{safety}$ (a margin for leakage through P and statistical deviations), as recommended by FE theory (Dodis et al., 2008).
- Domain strings ctx and meta (e.g., application/version/policy identifiers) prevent cross-domain linkage and enable rotation.

Confidentiality and linkability. Published P does not reveal R by definition (Dodis et al., 2008), but it can facilitate record linkability. To reduce risks: (i) include a random seed/version in P (different instances yield different P); (ii) apply domain separation in KDF; (iii) where possible, store LSH signatures instead of the full c in the index.

Fault tolerance. Observed tails of intra-BER are compensated by a margin in t and by the code choice; under degraded conditions (strong distortions) a record may fail to reconstruct (FRR). This is controlled by selecting n, t and by the profile of admissible transforms $\mathcal{T}$.

3.3 Overall architecture

Image x	$\xrightarrow{e(x)}$	embedding $\to$ binarization $\to c(x)$
		$\downarrow$ Gen (FE) $\Rightarrow (R, P)$
		$\downarrow$ KDF + HMAC/SHA-3 $\Rightarrow T$
Stored in DB	:	index: c (or LSH), public: P, cryptographic tag: T
Query x'	$\xrightarrow{c(x')}$	Hamming search $\Rightarrow$ candidates $\Rightarrow$ Rep$(c', P) \Rightarrow R' \Rightarrow T'$ compare with T

A reference implementation is available in a public repository: https://github.com/d-yacenko/Article14.git.

3.4 Limitations and practical notes

- The perceptual part remains vulnerable to targeted (adversarial) manipulations; tag security depends on FE+primitive, not on an "avalanche" property of the embedding (Struppek et al., 2021).
- For exact byte-level identity one may additionally store a classical hash (e.g., SHA3-256 of a canonicalized file); it is not used for search, but provides strict equality checks.
- Debugging/compatibility: fix model/library versions; use domain strings and version parameters (A, b) and the FE instance.

Thus, the proposed construction preserves similarity search at the perceptual-code level and adds on top a cryptographic verification tag with guarantees grounded in the fuzzy extractor and standard cryptographic primitives (Dodis et al., 2008; Menezes et al., 1996).

3.5 Use in two roles: search and steganography triage

The pipeline perceptual hash $\to$ fuzzy extractor (FE) addresses the main task of searching and verifying similar images under small distortions: the perceptual hash preserves similarity-aware retrieval, while the FE (Dodis et al., 2008) stabilizes a "noisy" bit fingerprint into a deterministic key/tag (Dang, 2012; Menezes et al., 1996).

Additional role: an indicator for steganography triage. We note an auxiliary signal that can be useful for investigations and security triage. If, for a pair of images, we simultaneously observe:

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

1. high perceptual similarity (small pHash distance) and successful reproduction in the FE (match=True), and
2. the files differ under byte-level comparison (including differences at the container/metadata level),

then such a pair can be assigned to the class "suspicious, requiring additional checks for steganography". The intuition is simple: steganographic methods often introduce small, visually imperceptible modifications (LSB/DCT/chunks) that do not destroy pHash and remain within the FE tolerance, but still change the byte representation. In the reference implementation https://github.com/d-yacenko/Article14.git, we provide an example: the original file images/0.07180100_1442390923_.jpg and the file images/0.07180100_1442390923_stegano.jpg that contains steganographic data (the string hello world!) are identified as similar by the proposed mechanism, yet differ even in size. This demonstrates the indicator.

Important limitation. This signal is a weak indicator: benign operations behave the same way (JPEG re-encoding, resize, crop, EXIF/ICC editing, auto-rotation, recompression, etc.). Therefore, this criterion is neither necessary nor sufficient to conclude the presence of steganography; it should be treated as a trigger for additional diagnostics.

Recommended triage pipeline. After the indicator triggers, perform lightweight automated checks:

1. Normalization and re-evaluation: strip metadata, convert images to a common format/size/quality, then recompute pHash and the FE; persistence of the match increases priority.
2. Container analysis: for JPEG, inventory APP/COM segments and tail after EOI; for PNG, check nonstandard ancillary chunks (iTXt/zTXt/custom) and anomalies in IDAT size.
3. Quick statistical tests: simple LSB/χ^2/RS tests for uncompressed/PNG, and a basic analysis of DCT coefficient distributions for JPEG (as probabilistic screening, not as proof).

This block does not change the cryptographic part; it only formalizes a "suspicion signal" that naturally arises from properties of pHash and the FE (Dodis et al., 2008; Dang, 2012; Menezes et al., 1996) and can be useful in practical security and digital forensics scenarios.

4 Information-theoretic and statistical analysis: source $c(x)$ and FE output

In this section we separate two levels: (1) the perceptual binary code $C = c(X) \in \{0,1\}^n$ as a source with "noise" (robust within class, separable between classes); (2) the cryptographic artifact $(R, P) = \mathrm{Gen}(C)$ and the derived tag T (via KDF/HMAC or SHA-3), for which cryptographic properties are formulated (Dodis et al., 2008; Menezes et al., 1996).

4.1 Properties of the perceptual source $C = c(X)$

Bit balance and weak correlations. For each bit i we estimate the frequency $p_i = \Pr[C_i = 1]$ and pairwise correlations $\mathrm{corr}(C_i, C_j)$ (and/or mutual information). The goal is $p_i \approx \frac{1}{2}$ and small $|\mathrm{corr}(C_i, C_j)|$ for $i \neq j$. This reduces the probability of random "sticking" and increases the effective entropy of the source, and also improves the subsequent Hamming-search quality (Gong & Lazebnik, 2011; Jégou et al., 2011; Ge et al., 2013; Datar et al., 2004).

Intra-BER and selecting radius t. For admissible transforms $T \in \mathcal{T}$ we measure

$$\mathrm{BER}_{\mathrm{intra}}(x, Tx) = \tfrac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{c_i(x) \neq c_i(Tx)\}.$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

We choose t so that, for example, $q_{0.95}(\mathrm{BER}_{\mathrm{intra}}) \cdot n \le t$—this ensures a high probability of successful reconstruction in the FE (low FRR).

Inter distribution and rarity of perceptual collisions. For dissimilar pairs (x, x') we build the distribution D_{inter} of $\mathrm{Ham}(c(x), c(x'))/n$. Ideally, the median is close to 0.5 and the left mass (near the candidate threshold r) is small. For a database of size M, the expected number of false candidates per query is

$$\mathbb{E}[\mathrm{FP}] \;\approx\; M \cdot p_{\le r}, \quad p_{\le r} = \Pr_{(x,x')\sim D_{\mathrm{inter}}}\big[\mathrm{Ham}(\cdot,\cdot) \le r\big].$$

This metric controls the accuracy and load of the search stage.

Min-entropy and leakage through public data. We estimate a lower bound on source min-entropy as

$$H_\infty(C) = -\log_2 \max_c \Pr[C = c],$$

in practice via code/cluster frequencies or blockwise lower bounds. Publishing helper data P (e.g., a linear-code syndrome of length $n-k$) reduces the available min-entropy by at most the leakage size; hence

$$\tilde{H}_\infty \;=\; H_\infty(C \,|\, P) \;\gtrsim\; H_\infty(C) - (n-k).$$

This quantity limits the maximum extractable secret $|R|$ and sets the security margin (Dodis et al., 2008).

What is <u>not</u> claimed at the level of $c(x)$. We do not require SAC/BIC in the strict sense and do not use the "birthday estimate" $2^{n/2}$: bits of C can be dependent/biased, and the code is vulnerable to targeted adversarial manipulations. Cryptographic properties are defined only on top of the FE and the tag primitive.

4.2 Properties of (R, P) and the cryptographic tag T over the FE

Secret extraction. The FE guarantees that, if $\mathrm{dist}_H(C, C') \le t$, reconstruction $\mathrm{Rep}(C', P) = R$ succeeds with high probability, and R is statistically close to uniform on $\{0,1\}^\ell$ given sufficient $\tilde{H}_\infty$ (closeness parameter ε is chosen in the regime $2^{-\lambda}$) (Dodis et al., 2008). We then derive a key $K = \mathrm{KDF}(R \,\|\, \mathrm{ctx})$.

Cryptographic tag. The tag $T = \mathrm{HMAC}_K(\mathrm{meta})$ (or $T = \text{SHA3-256}(K \,\|\, \mathrm{meta})$) inherits security (collision/second-preimage/preimage resistance) from the underlying primitive provided that R is unpredictable and domain separation ctx/meta is correct (Menezes et al., 1996). Thus:

- Collision resistance of the tag reduces to the security of the chosen primitive;
- Preimage/2nd-preimage reduce to the security of the primitive and the inability to predict R from P.

Note that tag equality means equality of the extracted secret (and thus "sufficient closeness" of codes), but not identity of the original pixels—this is an intentional FE property.

Compatibility with search. Candidate retrieval is performed using $c(x)$ (or its LSH signatures), while verification is done via FE→KDF→tag. Thus, both goals are preserved: fast perceptual search and cryptographically strong verification.

4.3 Section summary

In this section we considered two levels of the construction. At the perceptual-code level, we analyze the source $C = c(X)$ through bit balance, weak inter-bit correlations, intra-BER under admissible transforms, and the inter-distance distribution for dissimilar pairs. These

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

properties are intended to make the code suitable both for ANN search and as a noisy source for the fuzzy extractor.

At the FE/tag level, the goal is different: after applying the fuzzy extractor and KDF, we obtain a secret R and a tag T for which cryptographic guarantees are formulated in the standard way, i.e., security reduces to the underlying primitives together with the residual min-entropy budget after publishing P. Thus, the perceptual code itself is not treated as an independent cryptographic hash; rather, it provides a metrically robust source and index for search, while the cryptographic guarantees are assigned only to (R, P) and the derived tag T.

In this sense, the proposed architecture connects two objectives: an avalanche-like between-class separation at the perceptual level, and standard cryptographic verification at the tag level. The current evaluation should be read as preliminary evidence that this two-layer design is workable, rather than as a complete large-scale validation.

Applicability limitation. Perceptual/neural hashes (our code $c(x)$) are useful for similarity search and media deduplication, but they cannot be used as a full replacement for general-purpose cryptographic hash functions (integrity, signatures). At the level of $c(x)$, targeted transforms can lead to collisions or code preservation under noticeable changes of visual content (adversarial examples) (Struppek et al., 2021; Kotenko et al., 2023) . For cryptographic purposes, (preimage/second-preimage/collision) properties in our system are provided only after applying the fuzzy extractor (obtaining (R, P)) and subsequent cryptographic processing (KDF, HMAC/SHA-3), in line with common cryptographic practice.

Statistical tests (note). Practical checks include two groups of tests. (1) For the source $C = c(X)$: bit balance, inter-bit correlations/mutual information, distributions of $\mathrm{BER}_{\mathrm{intra}}$ and D_{inter}, and a lower-bound estimate of min-entropy $H_\infty(C)$. (2) For the extracted secret R: randomness test batteries (e.g., NIST STS; see survey-style discussions in (Doganaksoy et al., 2010)) and checks that no degradation occurs after publishing P (estimating $H_\infty(C \mid P)$ and the security budget). These tests do not replace a rigorous cryptanalysis, but help to detect systematic biases and leakages early.

5 Preliminary proof-of-concept experiments

5.1 Experimental setup

The current evaluation is a compact proof-of-concept intended to validate the mechanics of the proposed pipeline rather than to serve as a large-scale benchmark. We use a small demo collection of 140 images, including original images and a small number of benignly modified or stego-modified variants. The perceptual code length is $n = 512$ bits. For this collection, we evaluate: (i) intra-BER under admissible transformations, (ii) inter-image normalized Hamming distances for dissimilar pairs, and (iii) end-to-end retrieval / FE / tag-verification behavior.

5.2 Compact quantitative summary

5.3 Intra-BER and inter-distance distributions

The intra-BER histogram shows that admissible transformations usually remain within a moderate bit-error regime, which in turn determines the required error-correction capability for the fuzzy extractor. In the current experiment, the 95th percentile gives the practical estimate $t \approx 88$ bits for $n = 512$.

The inter-distance histogram shows that dissimilar-image pairs are shifted to the right relative to near-duplicate behavior, although the observed median remains below the idealized random-hash regime of 0.5. Accordingly, the current results should be interpreted as preliminary evidence of separability on a small demo dataset rather than as a final large-scale evaluation.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Metric	Value
Dataset size	140
Code length n	512
Intra-BER mean μ	0.1466
Intra-BER std σ	0.0260
Intra-BER quantile $q_{0.95}$	0.1719
Recommended FE radius t	88 bits
Inter-distance median	0.3066
Mass of inter-distances ≤ 0.3	0.4255
Inter-pair count	1389
Top-5 retrieved candidates verified	3 / 5
FE same-image reconstruction	success
FE perturbed-code reconstruction	success

Table 1: Compact summary of the preliminary proof-of-concept evaluation.

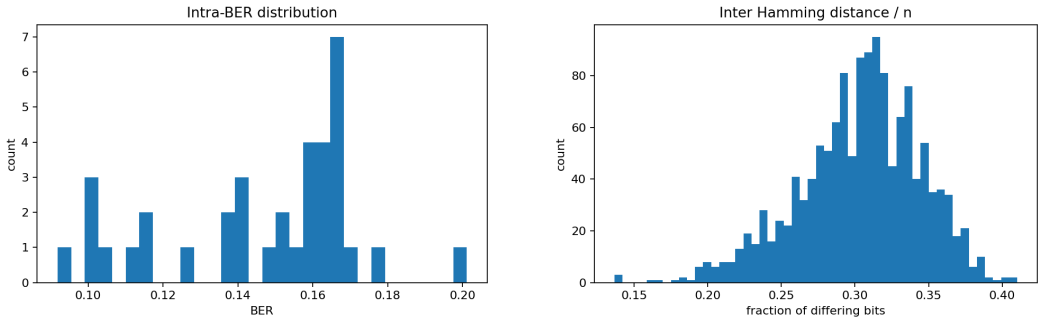


Figure 1: Left: histogram of intra-BER values for admissible transforms. Right: histogram of normalized Hamming distances for dissimilar image pairs.

5.4 Retrieval and verification behavior

In the retrieval experiment, the pipeline returns near candidates in Hamming space and then applies fuzzy-extractor reconstruction and tag verification. For the current top-5 retrieval example, 3 candidates pass FE/tag verification. This confirms that the end-to-end pipeline is operational as a proof of concept: perceptual retrieval is preserved, and cryptographic verification can be applied on top of retrieved candidates.

6 Conclusion

We presented a two-layer construction: (i) a perceptual binary code $c(x)$ obtained from a ResNet-18 embedding via projection and binarization, and (ii) a cryptographic layer based on a fuzzy extractor that produces a reproducible secret R and public data P, from which a cryptographic tag T is formed via KDF and HMAC/SHA-3. The current proof-of-concept experiments show that the pipeline is operational on a small demo dataset. For $n = 512$ bits and 140 images, we obtain an intra-BER mean of 0.1466, a 95th percentile of 0.1719 (suggesting a practical FE radius of about 88 bits), and an inter-distance median of 0.3066 for dissimilar pairs. These results indicate that near-duplicate and dissimilar pairs are separated to a useful extent for the retrieval stage, although the present evaluation remains preliminary and should not be interpreted as a large-scale benchmark.

At the level of (R, P) and the tag T, cryptographic guarantees (preimage / second-preimage / collision resistance) reduce to the security of the employed primitives and the estimated source min-entropy after publishing P. At the same time, the perceptual code itself should not be regarded as a standalone cryptographic hash. Overall, the proposed architecture combines perceptual indexing with a cryptographic verification layer and is intended to

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

provide the practically desirable combination of two goals: fast search for similar images and a cryptographically meaningful verification tag.

Limitations remain important. The perceptual part is inherently vulnerable to targeted adversarial perturbations, and the current empirical validation is small-scale. Future work should strengthen min-entropy estimation, evaluate linkability through P, study larger datasets and stronger baselines, and calibrate error-correcting codes to the observed intra-BER tails.

References

Thomas W. Cusick. Boolean functions satisfying a higher order strict avalanche criterion. In Tor Helleseth (ed.), Advances in Cryptology — EUROCRYPT '93, volume 765 of Lecture Notes in Computer Science, pp. 102–117, Berlin, Heidelberg, 1994. Springer. doi: 10.1007/3-540-48285-7_9.

Quan Dang. Recommendation for applications using approved hash algorithms. Technical Report NIST SP 800-107 Rev.1, NIST, 2012. URL https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-107r1.pdf.

Mayur Datar, Piotr Indyk, Nina Immorlica, and Vahab S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the 20th Annual Symposium on Computational Geometry (SoCG), pp. 253–262, 2004. doi: 10.1145/997817.997857. URL https://doi.org/10.1145/997817.997857.

Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM Journal on Computing, 38(1):97–139, 2008. doi: 10.1137/060651380. URL https://doi.org/10.1137/060651380.

Ali Doganaksoy, Barış Ege, Onur Koçak, and Fatih Sulak. Cryptographic randomness testing of block ciphers and hash functions. Cryptology ePrint Archive, 2010(564), 2010. https://eprint.iacr.org/2010/564.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2946–2953, 2013. doi: 10.1109/CVPR.2013.379. URL https://doi.org/10.1109/CVPR.2013.379.

Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 817–824, 2011. doi: 10.1109/CVPR.2011.5995432. URL https://doi.org/10.1109/CVPR.2011.5995432.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(1):117–128, 2011. doi: 10.1109/TPAMI.2010.57. URL https://doi.org/10.1109/TPAMI.2010.57.

Igor Kotenko, Igor Saenko, Oleg Lauta, Nikita Vasiliev, and Dmitry Iatsenko. Attacks against machine learning systems: Analysis and gan-based approach to protection. In Proceedings of the International Conference "Intelligent Information Technologies for Industry" (IITI 2023), volume 777 of Lecture Notes in Networks and Systems, pp. 49–59. Springer, Cham, 2023. doi: 10.1007/978-3-031-43792-2_5.

Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Handbook of Applied Cryptography. CRC Press, 1996. URL https://cacr.uwaterloo.ca/hac/about/chap9.pdf.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1023 DOI: 10.66693/mathai.1023

Special Issue: MathAI 2026 Selected Papers

Received: 05 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Lukas Struppek, Dominik Hintersdorf, Daniel Neider, and Kristian Kersting. Learning to break deep perceptual hashing: The use case neuralhash. arXiv preprint arXiv:2111.06628, 2021.

A. F. Webster and S. E. Tavares. On the design of s-boxes. In Advances in Cryptology – CRYPTO '85, volume 218 of Lecture Notes in Computer Science, pp. 523–534. Springer, 1986.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

MATHEMATICS OF NATURAL INTELLIGENCE

Evgenii E. Vityaev[1],*

[1] *Sobolev institute of mathematics SB RAS, Novosibirsk, Russia*

vityaev@math.nsc.ru

ABSTRACT

In the process of evolution, the brain has achieved such perfection that artificial intelligence systems do not have and which needs its own mathematics. The concept of cognitome, introduced by the academician K.V. Anokhin, as the cognitive structure of the mind – a high–order structure of the brain and a neural hypernetwork, is considered as the basis for modeling. Consciousness then is a special form of dynamics in this hypernetwork – a large-scale integration of its cognitive elements. The cognitome, in turn, consists of interconnected COGs (cognitive groups of neurons) of two types – functional systems and cellular ensembles. K.V. Anokhin sees the task of the fundamental theory of the brain and mind in describing these structures, their origin, functions and processes in them. The paper presents mathematical models of these structures based on new mathematical results, as well as models of different cognitive processes in terms of these models. In addition, it is shown that these models can be derived based on a fairly general principle of the brain works: *the brain discovers all possible causal relationships in the external world and draws all possible conclusions from them*. Based on these results, the paper presents models of: "natural" classification; theory of functional brain systems by P.K. Anokhin; prototypical theory of categorization by E. Roche; theory of causal models by Bob Rehter; theory of consciousness as integrated information by G. Tononi.

1 INTRODUCTION

K.V. Anokhin offers the following approach to solving the "difficult problem of consciousness" and the "brain-main problem" of neuroscience Anokhin (2021). He argues that the success of understanding the nature of consciousness critically depends on the creation of a detailed neuroscientific theory of the carrier of conscious experience – what has been called "mind" for centuries. To define the mind as a cognitive structure, K.V. Anokhin introduces the concept of "cognitome" – a high-order structure of the brain neural hypernetwork. In that case a cognition is a special form of dynamics in this hypernetwork – a large-scale integration of its cognitive elements.

By "***natural intelligence***" we will mean a cognitive hypernetwork of the brain, consisting of interconnected COGs (COgnitive Groups of neurons representing elements of subjective experience) of two types – functional systems and cellular ensembles by D. Hebb Anokhin (2021); Vityaev (2021). K.V. Anokhin sees the task of the fundamental theory of the brain and mind in describing these systems structures, their origin, functions and processes in them Anokhin (2021); Vityaev (2021).

The purpose of this paper is to provide a mathematical description of these structures and thereby a mathematical model of the cognitome. We will first show that the basic properties and functions of these two types of COGs can be derived from a more general principle***: the brain detects all possible causal relationships in the external world and makes all possible conclusions from them***.

For this, we first analyze the concept of causality. Causality is a consequence of physical determinism: "for any isolated physical system, some of its state determines all subsequent states" Carnap (1971). In the philosophy of science, causality is reduced to prediction and explanation. "Causality means predictability... if all the previous situation is known, the event can be predicted ... if all the

*Corresponding author.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

facts and laws of nature associated with this event are given" Carnap (1971). It is clear that it is impossible to know all the facts, the number of which is potentially infinite, and all the laws. In addition, humans and animals learn the laws of the external world through training (inductive inference). Therefore, causality is reduced to prediction by inductive statistical inference (Section 1), when the prediction is derived from facts and statistical data with some probability.

In addition, cause-and-effect relationships in the form of statistically laws that are detected on real data or as a result of training, face the *problem of statistical ambiguity* – contradictory predictions can be derived from them (Section 1) Hempel (1965; 1968). To avoid this ambiguity, Hempel introduced ***the maximum specificity requirement*** (Section 1), which informally means that statistical laws should include as much available information as possible.

We solved ***the problem of statistical ambiguity*** and determined ***the most specific statistical laws*** for which it was proved (Theorem 1) that the inductive-statistical inference using them does not lead to contradictions Vityaev (2006b); Vityaev & Martinovich (2015); Vityaev & Odintsov (2019). To detect such laws, a special ***semantic probabilistic inference was developed***. In addition, we have developed ***a formal neuron model*** that satisfies the Hebb rule, which infers such laws and discovers the most specific statistical laws Vityaev (2015).

Probabilistic cause-and-effect relationships, reflecting the relationships between the properties of some object of the external world, loop on themselves, forming cellular ensembles and creating fixed points of cyclically mutually predicting attributes. These fixed points have a special meaning and reflect the ***"natural" classification*** of objects in the external world (Section 2). It was noted that the "natural" classes of animals or plants differ in a potentially infinite set of properties Mill (1983). Natural scientists who built "natural" classifications noted that the construction of a "natural" classification consists in "indication" – from an infinitely large number of features, you need to move to a limited number of them, which would replace all other features Smirnov (1938). This means that in "natural" classes, features are strongly correlated, for example, if there are 128 classes and the attributes are binary, then about 7 attributes may be independent "indicator" attributes among them, since $2^7 = 128$, and other attributes can be predicted from the values of these 7 attributes. We can choose various 7-10 attributes as "indicators" and then other attributes, which are potentially infinitely many, are also predicted from these selected attributes. Therefore, there can be an exponential number of causal relationships between the attributes of objects of the "natural" classes relative to the number of attributes.

We formalize the "natural" classification by generalizing the analysis of formal concepts Ganter (2003); Ganter & Wille (2003). Formal concepts can be defined as a fixed points of deterministic rules (with no exceptions) Ganter (2003). We generalize formal concepts to the probabilistic case, replacing deterministic rules with probabilistic maximally specific causal relationships and defining ***probabilistic formal concepts*** as a fixed points of these maximally specific rules Vityaev & Martynovich (2015); Vityaev & Martinovich (2015); Vityaev et al. (2012); Vityaev & Odintsov (2019). Since the inference based on the most specific causal relationships is consistent (Theorem 1), the resulting fixed points will also be consistent and will not contain both a sign and its negation, i.e. such a definition of probabilistic formal concepts is correct.

It can be shown that probabilistic formal concepts adequately formalize the "natural" classification, and the resulting "natural" classification satisfies all the requirements that natural scientists imposed on "natural" classifications Vityaev & Martynovich (2015).

In addition, this formalization (Section 3) define a ***"natural" concepts*** described in the works of Eleanor Rosch Rosch (1975; 1978). The definition of "natural" concepts is based on the categorization principle of Eleanor Rosch, which postulates the structure of the perceived world: "the perceived world is not an unstructured set of equally likely properties, on the contrary, objects of the perceived world have a highly correlated structure" Rosch (1978).

The highly correlated structure of the external world is also basis for ***integrated information***, defined by G. Tononi (Tononi (2010; 2012); Oizumi et al. (2014)) (Section 4). But G. Tononi does not have a model of the external world, and integrated information is considered as an internal property of the system of causal relationships that manifests itself in consciousness. It defines the ***concept*** notion as a system of causal relationships with maximally integrated information. Since it has no external world, it cannot say that concept reflects "natural" concepts or "natural" objects.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Thus, our formalization of "natural" classification using probabilistic formal concepts serves as a formalization of both "natural" concepts and integrated information. It is principally different from the formalization of the causal systems, based on Bayesian networks, as it was done in Rehder & Martin (2011), since Bayesian networks do not support cycles.

In our formalization Cognitom is a reflection of the hierarchy of "natural" classes of the external world, in which properties and objects of the class form fixed points in the form of the probabilistic formal concepts. These fixed points generate the most integrated information according to G. Tononi. The algorithm of "natural" classification is based on a certain criterion of maximum consistency of causal relationships in its mutual predictions, which is close in its meaning to the integrated information.

Formalization of the second type of COGs – functional systems, is based on the consideration of purposeful behavior, which is carried out by the developing conditional (causal) relationships between actions and their results. In sections 8-11, it is shown that these conditional relationships are sufficient for modeling functional systems and developing animats.

2 CAUSALITY

As already mentioned, causality is reduced to prediction and explanation Carnap (1971).

There are two models for predicting some statement G:

1. Deductive-Nomological (D-N), based on facts and deductive laws.
2. Inductive-Statistical (I-S), based on facts and probabilistic laws.

Deductive-nomological model can be represented by the following scheme:

$$\frac{\frac{L_1, \ldots, L_m}{C_1, \ldots, C_n}}{G}$$

1. $L_1, \ldots, L_m$ – a set of laws.
2. $C_1, \ldots, C_n$– a set of facts.
3. G – predicted sentence.
4. $L_1, \ldots, L_m, C_1, \ldots, C_n \vdash G$;
5. the set $L_1, \ldots, L_m, C_1, \ldots, C_n$ is consistent.
6. $L_1, \ldots, L_m \nvdash G$, $C_1, \ldots, C_n \nvdash G$;
7. $L_1, \ldots, L_m$ - set of laws contain only generality quantifiers. Set $C_1, \ldots, C_n$of facts – quantorless formulas.

Inductive-statistical model is similar to the previous one, with the difference that the conclusion is made with some probability, so property 7 is formulated differently:

7. the set $L_1, \ldots, L_m$ contains statistical laws. Set of facts $C_1, \ldots, C_n$– quantorless formulas.

When detecting laws based on real data***, the problem of statistical ambiguity arises***, which consists in the fact that in the process of learning (inductive inference) we can obtain probabilistic rules from which the contradiction may be derived.

A classic example of statistical ambiguity.

Let's assume that there are the following statements:

(L1) – 'almost all cases of streptococcus are quickly cured by penicillin injection';

(L2) – 'Almost always penicillin-resistant streptococcal infection is not cured after penicillin injection';

(C1) – ' Jane Jones has a strep infection'.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

(C2) – ' Jane Jones received penicillin injection';

(C3 – ' Jane Jones has a penicillin-resistant strep infection'.

Two contradictory statements can be deduced from this set of statements: one that explains why Jane Jones will recover quickly (E), and the other that explains why Jane Jones will not recover quickly (E).

Explanation 1		Explanation 2	
L1		L2	
C1, C2	[r]	C2, C3	[r]
E		$\neg E$	

The conditions of both explanations do not contradict each other, both of them can be simultaneously true. However, their conclusions contradict each other.

To avoid contradictions, Hempel Hempel (1968) introduced the ***requirement of maximum specificity*** for statistical laws:

The rule F ⇒ G is ***maximally specific*** in the state of knowledge K,

F ⇒ G, p(G;F)= r	
F(***a***)	[r]
G(***a***)	

if for every class H for which both of the following statements, $\forall \mathrm{x}(\mathrm{H}(\mathrm{x}) \Rightarrow \mathrm{F}(\mathrm{x}))$ and H(***a***), belong to K, there exists a statistical law p(G; H) = r' in K such that r = r'. The idea behind the maximum specificity requirement is that if F and H both contain object ***a***, and H is a subset of F, then H has more specific information about object ***a***, and, therefore, the law p(G;H) should be preferred to the law p(G;F), but the law p(G;H) must have the same probability as the law p(G;F). So any specification of the set of objects H can not change the probability r.

Despite the fact that Hempel gave a fairly precise informal definition of the maximum specificity requirement, neither he nor his followers proved that I-S inference, based on the most specific rules, is consistent.

We solved the problem of statistical ambiguity and defined the most specific statistical laws for which we prove (Theorem 1) that an inductive-statistical inference, based on such laws does not lead to contradictions Vityaev (2006b); Vityaev & Martynovich (2015); Vityaev & Martinovich (2015); Vityaev & Odintsov (2019).

3 WHAT IS A "NATURAL" CLASSIFICATION?

The highly correlated structure of the external world was revealed in two theories – "natural" classification and "natural" concepts. At the same time, "natural" classification refers to the structure of objects in the external world, and "natural" concepts, studied in cognitive science, refer to the perception of these "natural" objects.

The first sufficiently detailed analysis of the "natural" classification belongs to J.S. Mill (Mill (1983)). First, let's separate "artificial" classifications from "natural" ones by J.S. Mill:

1. artificial classifications - *"Let's take any attribute, and if some things have it, and others do not, then we can base the division of all things into two classes on it".*
2. *natural classifications - "But if we turn to a class "animal" or "plant", and if we look at what features the individuals embraced by a given class differ from those not included in it, we will find that in this respect some classes differ greatly from others. ... have such a large number of characteristics that they cannot be enumerated"* .

J. S. Mill defines the "natural" classification as follows: *"Natural groups ... are defined by features, ... taking into account not only the features that are absolutely common to all the objects included in the group, but the whole set of those features, of which all are found in the majority of these objects, and the majority – in all. Consequently, our concept of a class – the image that this class is repre-*

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

sented in our mind - is the concept of a certain pattern that has the majority of the characteristics of this class."

The problem of defining "natural" classifications is still discussed in the literature Nat (2013). However, from our point of view, there is no sufficiently adequate formalization of the "natural" classification.

4 What is a "natural" concepts?

In the introduction, the categorization principle of Eleanor Rosch was mentioned. Thus, basic objects are information-rich bundles of observed properties, that create categorization: "Categories can be viewed in terms of their clear cases if the perceiver places emphasis on the *correlational structure of perceived attributes* ... By prototypes of categories we have generally meant the clearest cases of category membership" Rosch (1975; 1978). Later, Eleanor Rosch's theory of "natural" concepts was called prototype theory.

In further research, it was found that models based on traits, similarities, and prototypes are not sufficient to describe "natural" concepts. It is necessary to take into account theoretical, causal, and ontological knowledge related to concept objects. For example, people not only know that birds have wings, can fly and build nests in trees, but also that birds build nests in trees because they can fly, and fly because they have wings.

Considering these studies, Bob Rehder put forward the causal-model theory, according to which: "people's intuitive theories about categories of objects consist of a model of the category in which both a category's features and the causal mechanisms among those features are explicitly represented" Rehder (2003). In the theory of causal models, the relation of an object to a category is no longer based on the set of features and proximity by features, but on the similarity of the generating causal mechanism: "Specifically, a to-be-classified object is considered a category member to the extent that its features were likely to have been generated by the category's causal laws, such that combinations of features that are likely to be produced by a category's causal mechanisms are viewed as good category members and those unlikely to be produced by those mechanisms are viewed as poor category members" Rehder (2003).

Bob Rehder used Bayesian networks to represent causal models Rehder & Martin (2011). However, they do not support cycles and therefore cannot model cyclic causal relationships. Our proposed formalization by the probabilistic formal concepts directly models cyclic causal relationships represented by fixed points of predictions based on causal relationships Vityaev (2006b); Vityaev et al. (2012); Vityaev & Martinovich (2015); Vityaev & Martynovich (2015).

5 Integrated Information Theory by G. Tononi

G. Tononi's theory of integrated information is also based on the highly correlated structure of the external world Tononi (2010; 2012); Oizumi et al. (2014). Integrated information considered by G. Tononi as a property of a system of cyclic causal relationships: "Indeed, a "snapshot" of the environment conveys little information unless it is interpreted in the context of a system whose complex causal structure, over a long history, has captured some of the causal structure of the world, i.e. long-range correlations in space and time" Tononi (2012).

G. Tononi describes the relationship of integrated information with reality as follows: "Cause-effect matching ... measures how well the integrated conceptual structure ... fits or "matches" the cause-effect structure of its environment", "... matching should increase when a system adapts to an environment having a rich, integrated causal structure. Moreover, an increase in matching will tend to be associated with an increase in information integration and thus with an increase in consciousness" Tononi (2010; 2012).

G. Tononi defines consciousness as a primary concept that has the following phenomenological properties: composition, information, integration, exclusion Tononi (2010; 2012); Oizumi et al. (2014). We present the formulations of these properties together with our interpretation of these properties (given in parentheses) from the point of view of the "natural" classification.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

1. composition – elementary mechanisms (causal interactions) can be combined into higher-order ones ("natural" object classes form a hierarchy).

2. information – only mechanisms that specify 'differences that make a difference' within a system "count" (only the system of "resonating" causal relationships forming the class is significant).

3. integration – only information irreducible to non-interdependent components counts (only a system of "resonant" causal relationships is significant, which is not reduced to information of individual components, indicating a perception of a highly correlated structure of a "natural" object).

4. exclusion – only maxima of integrated information count (only values of features that are maximally interconnected by causal relationships form an "image" or "prototype").

Unlike G. Tononi, we consider these properties not as internal properties of the system, but as the ability of the system to reflect the "natural" classification of objects of the external world, and consciousness – as the ability of a complex hierarchical reflection of the "natural" classification of the external world.

6 PROBABILISTIC FORMAL CONCEPTS

We present our formalization of the basic concepts: Cartwright definitions of probabilistic causality, semantic probabilistic inference, maximally specific causality, and probabilistic formal concepts. This section follows the works Vityaev (2006b); Vityaev & Martinovich (2015); Ganter (2003); Ganter & Wille (2003).

Definition 1. *A formal context* $\mathrm{K} = (\mathrm{G}, \mathrm{M}, \mathrm{I})$ is a triple, where G and M– arbitrary set of objects and attributes, and $I \subseteq G \times M$– is a binary relation that expresses the belonging of an attribute to an object.

In a formal context, the derived operators play a key role – they link subsets of objects and attributes of the context.

Definition 2. For $A \subseteq GandB \subseteq M$:

1. $A^{\uparrow} = \{m \in M \mid \forall g \in A, (g, m) \in I\}$
2. $B^{\downarrow} = \{g \in G \mid \forall m \in B, (g, m) \in I\}$

Definition 3. A pair (A, B) is a formal concept if $A^{\uparrow} = B$ and $B^{\downarrow} = A$.

In the framework of logic, we will consider only finite contexts.

Definition 4. For a context $\mathrm{K} = (\mathrm{G}, \mathrm{M}, \mathrm{I})$, we define Ω_K a context signature, that contains predicate symbols for each $\mathrm{m} \in \mathrm{M}$, $K \models m(x) \Leftrightarrow (x, m) \in I$.

Definition 5. For the signature Ω_K, we define the following variant of first-order logic:

1. X_K - set of variables;
2. At_K– set of atomic formulas (atoms) $\mathrm{m}(x)$, $m \in \Omega_K$, $x \in X_K$;
3. L_K – set of literals that includes all atoms $\mathrm{m}(\mathrm{t})$ and their negations $\neg m(t)$;
4. Φ_K – set of formulas defined inductively: a literal is a formula, and for any $\Phi, \Psi \in \Phi_K$ expressions $\Phi \wedge \Psi,\ \Phi \vee \Psi,\ \Phi \rightarrow \Psi,\ \neg\Phi$ is also a formula.

We define conjunction $\wedge L$ and negation $\neg L = \{\neg P \mid P \in L\}$ of a set of literals $L \subseteq \mathrm{L}_K$.

Definition 6. A single signature Ω_K element $\{\mathrm{g}\}$ forms the model K_g of this object. The truth of the formula ϕ on the model K_g is defined as $g \models \phi \Leftrightarrow K_g \models \phi$.

Denition 7. We define *a probability measure* μ on a set G in the Kolmogorov sense. Then we can define the probability measure on the set of formulas as:

$$\nu : \Phi_K \rightarrow [0, 1],\ \nu(\phi) = \mu(\{g \mid g \models \phi\}).$$

We assume that there are no non-essential objects in the context, such as $\mu(\{g\}) = 0,\ g \in G$.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Definition 8. Let $\{H_1, H_2, \ldots, H_k, C\} \in \mathrm{L}_K, where\ C \notin \{H_1, H_2, \ldots H_k\}, k \geq 0$.

1. *There is a relationship* $R = (H_1 \wedge H_2 \wedge \ldots \wedge H_k \to C)$;
2. *The premise* $R^{\leftarrow}$ of the relation R is a set of literals $\{\mathrm{H}_1, \mathrm{H}_2, ..., \mathrm{H}_k\}$;
3. *The conclusion* of the relationship is $R^{\rightarrow} = C$;
4. *The length* of the relationship is $|R^{\leftarrow}|$.

Definition 9. *The probability* η of the relation R – is a quantity

$$\eta(R) = \nu(R^{\rightarrow}|R^{\leftarrow}) = {}^{\nu(R^{\leftarrow} \wedge R^{\rightarrow})}\!/_{\nu(R^{\leftarrow})}.$$

If the denominator $\nu(R^{\leftarrow})$ of the ratio is 0, then the probability is undefined.

Definition 10. The relation R_1 is a sub-relation of the relation R_2, denoted as $R_1 \sqsubset R_2$, if $R_1^{\rightarrow} = R_2^{\rightarrow},\ R_1^{\leftarrow} \subset R_2^{\leftarrow}$.

Definition 11. The relation R_1 *refines* the relation R_2, denoted as $R_2 < R_1$, if $R_2 \sqsubset R_1$ and $\eta(R_1) > \eta(R_2)$.

Definition 12. The relation R is *a probabilistic causal relationship*, if for every $\tilde{R}$, $(\tilde{R} \sqsubset R) \Rightarrow (\tilde{R} < R)$.

The probabilistic causality given by Cartwright [20] with respect to a certain background can be formulated in the above terms as follows. If the premise $R^{\leftarrow}$ of the relation R is a set of literals $\{\mathrm{H}_1, \mathrm{H}_2, ..., \mathrm{H}_k\}$ and we consider this set as a background, then each literal of the premise is a probabilistic reason for the conclusion $R^{\rightarrow}$ with respect to this background, that is

$\nu(R^{\rightarrow}/R^{\leftarrow}) > \nu(R^{\rightarrow}/(R^{\leftarrow} \backslash H))$ for everyone $\mathrm{H} \in \{\mathrm{H}_1, \mathrm{H}_2, ..., \mathrm{H}_k\}$.

It is easy to see that this definition follows from definition 12.

Definition 13. *The strongest probabilistic causal relationship* is a relation R for which there is no such probabilistic causal relationship $\tilde{R}$ that $(\tilde{R} > R)$.

Definition 14. *The Semantic Probabilistic Inference* (SPI) of predictions of a certain literal C is a sequence of probabilistic causal relationships $R_0 < R_1 < R_2 ... < R_m$, $R_0^{\rightarrow} = R_1^{\rightarrow} = R_2^{\rightarrow} ... = R_m^{\rightarrow} = C$, $R_0^{\leftarrow} = \emptyset$, R_m – strongest probabilistic causal relationship.

Definition 15. The tree of *semantic probabilistic inference* Tree(C) of a certain literal C – is a collection of all SPI predictions of the literal C.

Definition 16. *The most specific causal relation* for predicting a certain C – is the strongest probabilistic causal relation of the Tree(C) with the maximum conditional probability.

Let us denote by MSCR the set of all maximally specific causal relations for predicting some literal C. By *the system of causal relations* we mean any subset $\mathcal{R} \subseteq \mathrm{MSCR}$.

Definition 17. Define *the prediction operator* for the system $\mathcal{R}$ as:

$$\Pi_{\mathcal{R}}(L) = L \cup \{C \mid \exists R \in \mathcal{R} : R^{\leftarrow} \subseteq L, R^{\rightarrow} = C\}.$$

Definition 18. By the *closure* of a set of literals L we mean the smallest fixed point of the prediction operator containing L:

$$\Pi_{\mathcal{R}}^{\infty}(L) = \bigcup_{k \in \mathrm{N}} \Pi_{\mathcal{R}}^{k}(L).$$

A set of literals L *is consistent* if it does not contain both the atom C and its negation $\neg C$. A set of literals L is *compatible* if $\nu(\wedge L) \neq 0$.

Theorem 1. If L is compatible, then $\Pi_{\mathcal{R}}(L)$ compatible and consistent for any system $\mathcal{R}$.

Proof of Theorem 1. Below are the three lemmas necessary to prove Theorem 1, and then the proof of the theorem itself.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Lemma 1. Any probabilistic causal relationship $\mathrm{C} = (A_1\&...\&A_k \Rightarrow A_0)$ belongs to some semantic probabilistic inference of the letters A_0, and, consequently, to the tree of semantic probabilistic inference of the letter A_0.

Proof. For probabilistic causal relationship $\mathrm{C} = (A_1\&...\&A_k \Rightarrow A_0)$, $k \geq 1$ we find a sub-relationship that is a probabilistic causal relationship. This always exists, because the rule $\mathrm{C} = (\Rightarrow A_0)$ is a probabilistic causal relationship. The conditional probability of this sub-relationship will be strictly less than the conditional probability of the causation itself. We add it as a previous rule for semantic probabilistic inference and continue the procedure.

Lemma 2. For any rule $\mathrm{C} = (\mathrm{A}_1\&...\&\mathrm{A}_\mathrm{k} \Rightarrow \mathrm{A}_0)$, $k \geq 0$,$\mathrm{A}_0 \notin \{\mathrm{A}_1\&...\&\mathrm{A}_\mathrm{k}\}$, $\eta(\mathrm{A}_1\&...\&\mathrm{A}_\mathrm{k}) > 0$ there is a probabilistic causal relationship $C' = (B_1\&...\&B_{k'} \Rightarrow \mathrm{A}_0)$, $B_1\&...\&B_{k'} \subseteq \mathrm{A}_1\&...\&\mathrm{A}_\mathrm{k}$, $k' \leq k$, for which $\mu(C') \geq \mu(C)$.

Proof. A rule $\mathrm{C} = (\mathrm{A}_1\&...\&\mathrm{A}_\mathrm{k} \Rightarrow \mathrm{A}_0)$ is either a probabilistic causal relationship and then it is the desired one, or, by virtue of the definition of a probabilistic causal relationship, there is sub-rule $(\tilde{R}\sqsubset R)$, $\tilde{R} = (P_1\& \ldots \&P_{k'} \Rightarrow \mathrm{A}_0)$, $k' \geq 0$, $\{P_1, \ldots, P_{k'}\} \subset \{A_1, \ldots, A_k\}$, $k' < k$ for which $\mu(\tilde{R}) \geq \mu(A)$. Similarly for a rule R', either it is a probabilistic causal relationship, or there is a sub-rule with similar properties for it, etc. Since the rule $\mathrm{C} = (\Rightarrow \mathrm{A}_0)$is a probabilistic causal relationship, the process will stop.

Lemma 3. If the inequality $\eta(G/\bar{A}\&\neg\bar{B}) > \eta(G/\bar{A})$ is true for the rules $\mathrm{A} = (\bar{\mathrm{A}} \Rightarrow \mathrm{G})$, $\mathrm{B} = (\bar{\mathrm{B}} \Rightarrow \neg\mathrm{G})$, $\bar{\mathrm{A}} = \mathrm{A}_1\&...\&A_k$, $\bar{\mathrm{B}} = \mathrm{B}_1\&...\&B_m$, $\eta(\bar{\mathrm{A}}\&\neg\bar{\mathrm{B}}) > 0$, $k \geq 0$, $m > 0$, then there exists a rule with a strictly higher conditional probability than the rule A.

Proof. Let's rewrite the conditional probability

$$\eta(G/\bar{A}\&\neg\bar{B}) = \eta(G/\bar{A}\&(\neg\mathrm{B}_1 \vee ... \vee \neg\mathrm{B}_m)).$$

Let us represent a disjunction $\neg\mathrm{B}_1 \vee ... \vee \neg\mathrm{B}_m$ as a disjunction of conjunctions

$$\bigvee_{i=(0,...,0)}^{i=(1,...1,0)}(\mathrm{B}_1^{i_1}\&...\&\mathrm{B}_m^{i_m}),\ i = (i_1, ..., i_m),\ i_1, ..., i_m \in \{0, 1\},$$

where zero means the presence of negation in the corresponding atom, and one – means the absence of negation. The disjunction does not include the tuple (1,...,1) corresponding to the conjunction $\mathrm{B}_1\&...\&\mathrm{B}_m$.

Then the conditional probability $\eta(G/\bar{A}\&(\neg\mathrm{B}_1 \vee ... \vee \neg\mathrm{B}_m))$ is rewritten as $\eta\left(G/\bigvee_{i=(0,...,0)}^{i=(1,...1,0)}(\bar{A}\&\mathrm{B}_1^{i_1}\&...\&\mathrm{B}_m^{i_m})\right)$.

We prove that if $\eta(G/\bar{A}\&\neg\bar{B}) > \eta(G/\bar{A})$, then one of the inequalities also holds

$$\eta(G/\bar{A}\&B_1^{i_1}\&...\&B_m^{i_m}) > \eta(G/\bar{A}), (i_1, ..., i_m) \neq (1, ..., 1).$$

To the contrary, assume that all inequalities are met simultaneously

$$\eta(G/\bar{A}\&B_1^{i_1}\&...\&B_m^{i_m}) \leq \eta(G/\bar{A}), (i_1, ..., i_m) \neq (1, ..., 1)$$

in cases where $\eta(\bar{A}\&B_1^{i_1}\&...\&B_m^{i_m}) > 0$.

Since $\eta(\bar{A}\&\neg\bar{B}) > 0$, there are cases when $\eta(\bar{A}\&B_1^{i_1}\&...\&B_m^{i_m}) > 0$.

Then

$$\eta(G\&\bar{A}\&B_1^{i_1}\&...\&B_m^{i_m}) \leq \eta(G/\bar{A})\eta(\bar{A}\&B_1^{i_1}\&...\&B_m^{i_m}), (i_1, ..., i_m) \neq (1, ..., 1),$$

$$\eta\left(G/\bigvee_{i=(0,...,0)}^{i=(1,...1,0)}(\bar{A}\&\mathrm{B}_1^{i_1}\&...\&\mathrm{B}_m^{i_m})\right) = \frac{\eta\left(\bigvee_{i=(0,...,0)}^{i=(1,...1,0)}(G\&\bar{A}\&\mathrm{B}_1^{i_1}\&...\&\mathrm{B}_m^{i_m})\right)}{\eta\left(\bigvee_{i=(0,...,0)}^{i=(1,...1,0)}(\bar{A}\&\mathrm{B}_1^{i_1}\&...\&\mathrm{B}_m^{i_m})\right)} =$$

$$\frac{\sum_{i=(0,...,0)}^{i=(1,...1,0)}\eta(G\&\bar{A}\&\mathrm{B}_1^{i_1}\&...\&\mathrm{B}_m^{i_m})}{\sum_{i=(0,...,0)}^{i=(1,...1,0)}\eta(\bar{A}\&\mathrm{B}_1^{i_1}\&...\&\mathrm{B}_m^{i_m})} \leq \frac{\eta(G/\bar{A})\sum_{i=(0,...,0)}^{i=(1,...1,0)}\eta(\bar{A}\&\mathrm{B}_1^{i_1}\&...\&\mathrm{B}_m^{i_m})}{\sum_{i=(0,...,0)}^{i=(1,...1,0)}\eta(\bar{A}\&\mathrm{B}_1^{i_1}\&...\&\mathrm{B}_m^{i_m})} = \eta(G/\bar{A}),$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

which contradicts the inequality $\eta(G/\bar{A}\&\neg\bar{B}) > \eta(G/\bar{A})$. Therefore, our assumption is not true and there is a rule of the form

$$\bar{A}\&B_1^{i_1}\&...\&B_m^{i_m} \Rightarrow G, (i_1, ..., i_m) \neq (1, ..., 1)$$

having a strictly higher conditional probability than A .

Proof of the theorem.

First we prove that each time a rule from $\mathrm{P} \subset \mathrm{MSCR}$ is applied, we again get a compatible set of letters. Suppose, on the contrary, that applying of a certain rule $\mathrm{A} = (\mathrm{A}_1\&...\&A_k \Rightarrow \mathrm{G})$, $\{\mathrm{A}_1, ..., A_k\} \subset L$, $k > 1$ to a set of letters $L = \{L_1, ..., L_n\}$ outputs the letter G, for which $\nu(\mathrm{L}_1\&...\&L_n\&G) = 0$.

Since for rules MSCR the inequalities $\eta(\mathrm{G}/\mathrm{A}_1\&...\&A_k) > \nu(\mathrm{G})$, $\nu(\mathrm{A}_1\&...\&A_k) > 0$, $\nu(\mathrm{G}) > 0$ are satisfied, then

$$\nu(\mathrm{G}\&\mathrm{A}_1\&...\&A_k) > \nu(\mathrm{G})\nu(\mathrm{A}_1\&...\&A_k) > 0.$$

Add negations of letters $\{B_1, ..., B_t\} = \{\mathrm{L}_1\&...\&L_n\}\backslash\{\mathrm{A}_1, ..., A_k\}$ to the rule A, then we get the rule $(\mathrm{A}_1\&...\&A_k\&\neg(B_1\&...\&B_t) \Rightarrow \mathrm{G})$.

Let's denote $\bar{\mathrm{A}} = \mathrm{A}_1\&...\&A_k$, $\bar{B} = B_1\&...\&B_t$, $\bar{\mathrm{L}} = \mathrm{L}_1\&...\&L_n$.

By the assumption $\nu(\mathrm{L}_1\&...\&L_n\&G) = 0$ of and $\nu(\bar{\mathrm{A}}\&\bar{B}) = \nu(\bar{\mathrm{L}}) > 0$. Let us prove that in this case $\nu(\bar{\mathrm{A}}\&\neg\bar{B}) > 0$. Suppose the opposite that $\nu(\bar{\mathrm{A}}\&\neg\bar{B}) = 0$, then $\nu(\mathrm{G}\&\bar{\mathrm{A}}\&\neg\bar{B}) \leq \nu(\bar{\mathrm{A}}\&\neg\bar{B}) = 0$, where it follow that

$$0 = \nu(\bar{\mathrm{L}}\&G) = \nu(\mathrm{G}\&\bar{\mathrm{A}}\&\bar{B}) = \nu(\mathrm{G}\&\bar{\mathrm{A}}) - \nu(\mathrm{G}\&\bar{\mathrm{A}}\&\neg\bar{B}) = \nu(\mathrm{G}\&\bar{\mathrm{A}}) > 0.$$

We got a contradiction. Then

$$\nu(\mathrm{G}/\bar{\mathrm{A}}\&\neg\bar{B}) = \frac{\nu(\mathrm{G}\&\bar{\mathrm{A}}\&\neg\bar{B})}{\nu(\bar{\mathrm{A}}\&\neg\bar{B})} = \frac{\nu(\mathrm{G}\&\bar{\mathrm{A}}) - \nu(\mathrm{G}\&\bar{\mathrm{A}}\&\bar{B})}{\nu(\bar{\mathrm{A}}) - \nu(\bar{\mathrm{A}}\&\bar{B})} =$$
$$\frac{\nu(\mathrm{G}\&\bar{\mathrm{A}}) - \nu(\mathrm{G}\&\bar{\mathrm{L}})}{\nu(\bar{\mathrm{A}}) - \nu(\bar{\mathrm{A}}\&\bar{B})} = \frac{\nu(\mathrm{G}\&\bar{\mathrm{A}})}{\nu(\bar{\mathrm{A}}) - \nu(\bar{\mathrm{A}}\&\bar{B})} > \frac{\nu(\mathrm{G}\&\bar{\mathrm{A}})}{\nu(\bar{\mathrm{A}})} = \nu(G/\mathrm{A}_1\&...\&A_k).$$

Then, by lemmas 2, 4, 5, we get that there is a probabilistic causal relationship with a higher conditional probability than the rule A, which contradicts the maximum specificity of the rule A.

Since the set of letters $\mathrm{L}_1, ..., L_n, G$ is compatible and $\nu(\mathrm{L}_1\&...\&L_n\&G) > 0$, then it is consistent, since if it contained both letters G and $\neg G$, then its probability would be zero.

Definition 19. *A probabilistic formal concept* on the context K – is a pair (A, B) satisfying the following conditions:

$$\Pi_{\mathcal{R}}^{\infty}(B) = B,\ A = \bigcup_{\Pi_{\mathcal{R}}^{\infty}(C)=B} C^{\downarrow}.$$

The definition of a set A is based on the following theorem linking probabilistic and standard formal concepts on the context K.

Theorem 2. Let $\mathrm{K} = (\mathrm{G}, \mathrm{M}, \mathrm{I})$ be a formal context, then:

1. If (A,B) is a formal concept on K, then there exists a probabilistic formal concept (S,T) on K such that $A \subseteq S$.
2. If (S,T) is a probabilistic formal concept on K, then there exists a family $\mathcal{C}$ of formal concepts on K such that

$$\forall (A, B) \in \mathcal{C}\ (\Pi_{\mathcal{R}}^{\infty}(B) = T),\ S = \bigcup_{(A,B)\in\mathcal{C}} A.$$

Proof of the theorem:

1. Put $T = \Pi_{\mathcal{R}}^{\infty}(B)$ and $S = \bigcup_{\Pi_{\mathcal{R}}^{\infty}(C)=B} C^{\downarrow}$. Then it is obvious that $B \subseteq T$ and $A \subseteq S$.

2. Define $P = \{Q|\Pi_{\mathcal{R}}^{\infty}(Q) = T, Q \subseteq M\}$. By P we construct a set of strict concepts $\mathcal{C} = \{(Q^{\downarrow}, Q^{\downarrow\uparrow})|Q \in P\}$. Then, if we define $S = \bigcup_{(A,B)\in\mathcal{C}} A$, then the conditions of the theorem are satisfied.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

7 CAUSALITY AND THE FORMAL NEURON MODEL

We present a formal model of a neuron that discover maximally specific conditional connections.

By *the information* received at the brain's "input" we will understand all the stimulation perceived by the brain: motivational, situational, trigger, sanctioning, reverse afferentation about actions performed, arriving at the "input" via collaterals, and so on. From the ecological theory of perception by J. Gibson Gibson (1988), it follows that information can be understood as any characteristic of the energy flow of light, sound, etc., coming to the "input" of the brain.

We define the information transmitted by the excitation of a certain nerve fiber to the synapses of a neuron by single predicates

$$P_j^i(a) \Leftrightarrow (x_i(a) = x_{ij}), i = 1, ..., n;\ j = 1, ..., n_i,$$

where$x_i(a)$- information, and x_{ij}– its value in the current situation (on the object) ***a***. If information is transmitted to the excitatory synapse, then it is perceived by the neuron as the truth of the predicate $P_j^i(a)$, if to the inhibitory synapse, then as the falsity $\neg P_j^i(a)$ of the predicate. We define the excitation of a neuron in situation ***a*** and the transmission of this excitation to the axon of the neuron by a predicate $P_0(a)$. If the neuron is inhibited in situation ***a***, then we define this situation as predicting the negation of the predicate $\neg P_0(a)$. Predicates $P_j^i(a)$, $P_0(a)$ and their negations $\neg P_j^i(a)$, $\neg P_0(a)$ are literals (atomic statements or their negations), which we denote as $a, b, c, ... \in L$, where L is the set of all literals in the dictionary $\{P_0\} \cup \{P_j^i\},\ i = 1, ..., n;\ j = 1, ..., n_i$.

Each neuron has its own receptive field that excites it unconditionally. The initial (before any training) semantics of the predicate P_0 is this receptive field.

We assume that the formation of conditional connections at the neuron level occurs according to the Hebb rule Hebb (1949). We formalize the Hebb rule using semantic probabilistic inference (definition 14, Fig. 1), which is fundamentally different from other formalizations in that it reveals maximally specific causal relationships.

Therefore, a neuron in the process of semantic probabilistic inference detects a set of $\{\mathrm{R}\}$ maximally specific causal relationships of the form:

$$R = (a_1 \& ... \& a_k \Rightarrow b), a_1, ..., a_k, b \in L,$$

where $a_1, ..., a_k$ are the predicates coming to the synapses of the dendrites of the neuron, and b is the predicate $P_0(a)$ or $\neg P_0(a)$ axon of the neuron. The tree of semantic probabilistic inference in Fig. 1 and the structure of the neuron, presented in Fig. 2, are similar, so it is easy to imagine that a neuron implements such inference.

Let's describe the properties of the resulting formal neuron model:

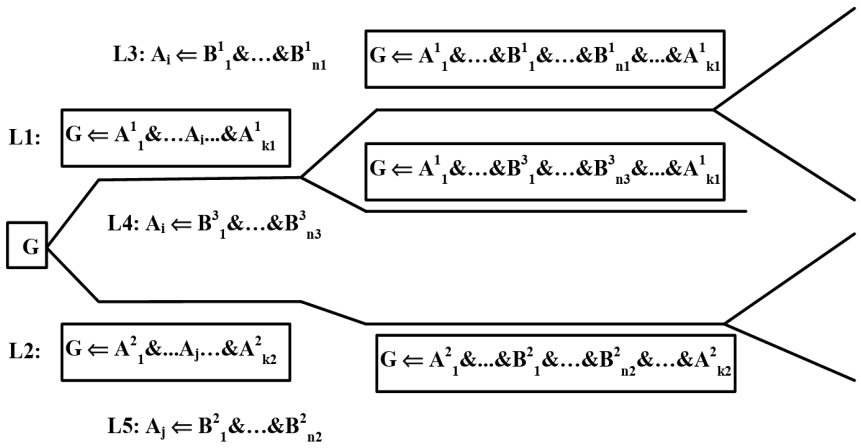


Figure 1: Semantic probabilistic inference tree

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

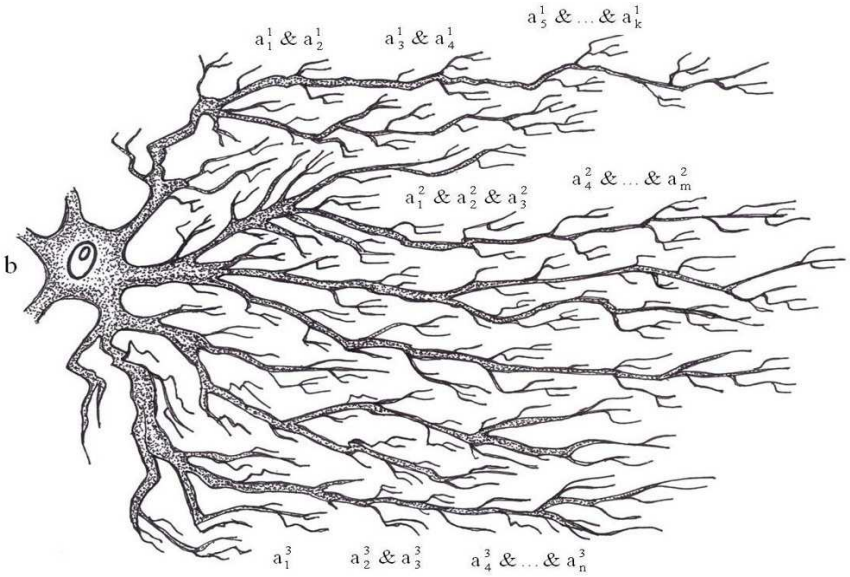


Figure 2: Formal model of neuron

1. The neuron performs "closure of conditional connections". When conditional stimuli are detected that allow predicting with some probability the excitation/inhibition of a neuron, conditional connectionis are formed that constitute the corresponding rule. When new stimuli are detected that allow predicting the excitation/inhibition of a neuron with even greater probability, it attaches them to this conditional connection.
2. The neuron is activated or inhibited according to the most probable rules. This is confirmed by the fact that in the process of discovering conditional connections, as well as when closing conditional connections at the neuron level, the speed of the neuron's response to a conditional signal is higher, the higher the probability of a conditional connection. Since the most specific rules that take into account all available information are at the same time the most probable, the prediction (neuron excitation) is carried out according to them;
3. The prediction made by neuron according to the most specific rules is consistent. Therefore, the neuron learns to predict without contradictions – either its excitatory, or inhibitory maximally specific rules are triggered, but not simultaneously.
4. Figure 2 shows several semantic probabilistic inferences made by a neuron. For example, a conditional connection $(b \Leftarrow a_1^1 \& a_2^1)$ is enhanced by new stimuli $a_3^1 \& a_4^1$ providing the connection $(b \Leftarrow a_1^1 \& a_2^1 \& a_3^1 \& a_4^1)$, if the stimuli $a_3^1 \& a_4^1$increase the conditional probability of predicting the firing of a neuron b.

Semantic probabilistic inference and the Discovery software system that implements it as a neuron model, have been successfully used to solve a number of applied problems Kovalerchuk & Vityaev (2000); Vityaev (2006a); Vityaev & Kovalerchuk (2017).

8 TASK CONCEPT AND THE FOUNDATIONS OF MATHEMATICS

We present an analysis of the ***desire*** concept given in Ershov & Samokhvalov (1984). Despite the generality of the above arguments, the mathematical result and revision of the foundations of mathematics obtained on the basis of this analysis in Ershov & Samokhvalov (1984) is its immediate consequence.

"I'm thirsty – what does it mean? There is, of course, no mistake in thinking that the words "I am thirsty" simply mean this, where it is a certain state of consciousness that I am experiencing now and that I call thirst. But then a new question arises: how does the feeling of thirst (wishing) relate to actual drinking (satisfying wishing)? How do I know that drinking can satisfy your thirst? Does

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

the experience of thirst itself contain a consciousness of how this thirst can be satisfied? ***... To know a desire does not mean to know what is desired, but to know the ability to recognize what is desired****as soon as the opportunity presents itself. In other words, you understand a desire ... only when you have matched that desire with a sense of confidence that you will be able to recognize any future state of consciousness in a convincing and unmistakable way as a state of desire satisfaction or a state of dissatisfaction... Although ... I don't necessarily know how this quenching will be achieved. In my past experience, I expect it to be water, but maybe some pill will also quench my thirst"*.

This argument allows us to clarify the ***tasks*** concept. "We understand a problem only when we compare it with a reasonable sense of confidence that we will be able to recognize any state of our consciousness in a convincing and unmistakable way as such when a solution is found, or as such when a solution is not found" Ershov & Samokhvalov (1984). Note that if the latter condition is not met, then the problem does not require a solution, since then any state of consciousness can be considered a solution.

Suppose we have some text. Is it a "convincing and error-free" presentation of the solved task? In mathematical theories, it is generally accepted that a "reasonable sense of confidence" that the statement of the solution of the task indeed its solution arises when this statement is a proof of the task. The proof provides a formal criterion for having a task solution. Let us assume that our states of consciousness, together with the evidence, can be formalized within the framework of some formal system S. Let us ask ourselves: does this formal system allow us to determine, by means of the formal system S itself, whether it is a proof of the solution of the problem or not? If such a formal system exists, it means that it can serve as a formal model for setting and solving mathematical problems. This question was analyzed in Ershov & Samokhvalov (1984; 2007) and it was proved that only in "weak" formal systems, in which Godel's incompleteness theorem does not pass, we can always determine by means of the formal system itself whether a certain text is a proof of the solution of a certain problem or not.

This result allowed its authors to formulate a new approach to the foundations of mathematics, consisting in a radical change in Hilbert's program for the justification of mathematics. "As you know, Hilbert believed that, generally speaking, not all statements of any mathematical theory make sense. At the same time, he implicitly assumed that the division of the set of all statements of the theory under consideration into meaningful ("real") and meaningless ("ideal") is completely determined by the type of statements themselves and, therefore, is fixed for all theories with the same syntax and signature. According to the new paradigm, this division into meaningful and meaningless statements depends not only on the syntax and signature of the theory under consideration, but also on the class of problems that this theory is intended to deal with. From this point of view, the same theory as a mathematical calculus will have different sets of meaningful statements if it is designed to handle different classes of problems. In other words, mathematical theory is considered simply as a "reservoir" for poorer formal systems, which are separately "extracted" from the whole theory depending on a particular problem at hand. By itself, regardless of possible tasks ... the theory has no practical significance, and therefore the question of whether it is contradictory in general or not is of no independent interest" Ershov & Samokhvalov (1984).

But we are not only interested in mathematical problems. We reformulate the concept of a task so as not to appeal to states of consciousness. We will say that ***a task is meaningful*** if and only if we have ***a criterion for solving the task***, in the sense that for each proposed solution we are always able to determine whether it is a solution or not. Tasks in this sense arise not only in mathematics, but also in many other areas, and therefore in all these cases it should be borne in mind that a criterion for task solving is always necessary.

9 Purpose and purposeful activity

Desire is active – it forces the body to show its activity in aimed at satisfying its desire. Then the concept of ***goal*** arises. You cannot achieve a goal without having a criterion for achieving it; otherwise, one could always assume that the goal has already been achieved. The criterion for achieving the goal is the satisfaction of the desire. The concept of a goal is more general than the concept of a task – the goal of the task is its solution and the criterion for achieving this goal is the criterion of task solution.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Defining a goal does not make sense without a criterion for achieving it, because we need to ascertain that the criterion is not currently met, and therefore, it makes sense to set a ***goal as something we do not have now but wish to achieve***. This definition of the goal allows us to define ***the result*** of achieving the goal, as all that we get when the criterion is met and the goal is achieved (desire satisfaction). There is the following relationship between the concepts of goal and result: the result is obtained when the goal is achieved and the achievement criterion is "triggered". But when a goal is set, we have a goal, but we don't have a result.

The definition of a goal is paradoxical, since the activity/activity to meet a certain criterion does not fundamentally imply knowledge about what and how to achieve the goal. You can set a goal without defining how, what and when to achieve it. This paradoxical nature of the goal concept is called ***the goal paradox***. As we will see from functional systems theory, brain activity in goal-directed behavior is constantly focused on resolving the goal paradox and determining how, what and when the goal can be achieved.

The action is always purposeful. If there is no goal for the action, then it is not clear when (or how) it should end. The meaning of activity is to change the current state in order to achieve something. Purposeful activity is aimed at satisfying a certain need (desire) of the body.

10 Functional Systems Theory

Functional Systems Theory is only one in which the concepts of goal, result and purposeful activity are central and where the physiological mechanisms of goal, result and purposeful activity are analyzed. Functional Systems Theory (TFS) is a theory of how the brain works as a system for achieving goals and resolving the goal paradox. Therefore, we will present the theory of functional systems as a theory of brain resolution of the goal paradox, which describes how the brain determines: how, what and when the goal can be achieved.

P.K. Anokhin wrote: "Perhapsone of the most dramatic moments in the history of studying the brain as an integrative education is the fixation of attention on the action itself, and not on its results ... we can assume that the result of the "grasping reflex" will not be grasping itself as an action, but that set of afferent stimuli that corresponds to the signs of the "grasped" object" Anokhin (1978; 1974; 1984)". The "set of afferent stimuli" is the criterion for achieving the goal in the TFS. It should be noted that this dramatic moment persists to nowadays.

Architecture of functional systems. According to P.K. Anokhin, the central mechanisms of functional systems that provide purposeful behavioral acts have the same architecture.

Afferent synthesis. The initial stage of a behavioral act of any degree of complexity consists of afferent synthesis, which includes the synthesis of motivational arousal, memory, situational and trigger afferentation.

Motivational arousal. Goal setting is carried out by the need that has arisen, which is transformed into motivational arousal.

Memory. Motivational arousal "extracts from memory" all possible ways to achieve a goal, as well as the entire sequence and hierarchy of results that must be obtained in order to achieve the goal in some specific way.

Situational afferentation. When a engram is recorded in memory, the environment in which the result was obtained is also recorded. This environment is recorded as the necessary conditions, along with the motivation required to achieve results. Therefore, motivational arousal in a given situation "extracts from memory" only those ways of achieving the goal that are possible in this situation. Thus, situational afferentation, when interacting with the experience extracted from memory, determines what and how can be done in this environment to achieve the goal.

Trigger afferentation. In its meaning, it is also a situational afferentation, only it is not related to the stimuli of the situation, but to the time and place of achieving the goal. Therefore, trigger afferentation answers the question: when and where can the result be achieved?

Thus, at the stage of afferent synthesis, the goal paradox is largely resolved, and it is determined what, how, where, and when something can be done to achieve the goal.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

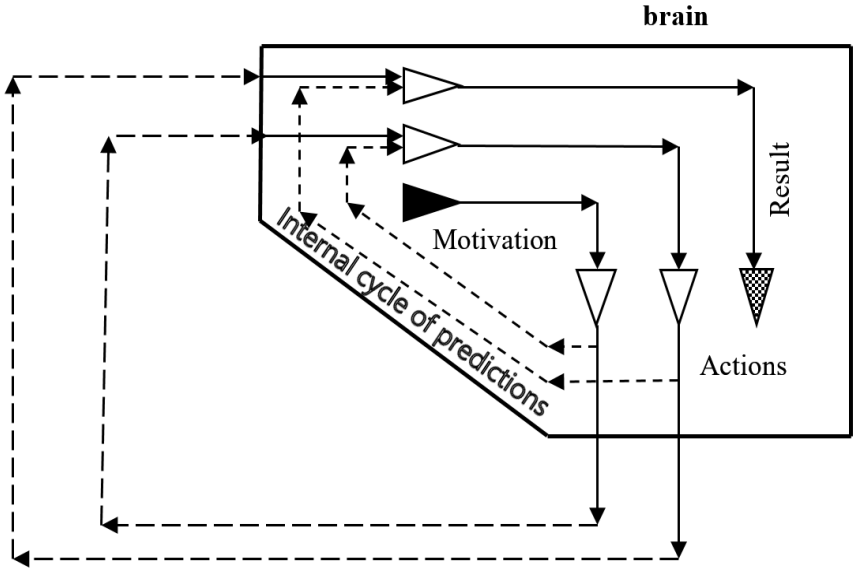


Figure 3: Formation of the inner contour of predictions

Decision making. At the stage of afferent synthesis by motivational arousal it can be extracted from memory (in a given situation) several ways to achieve a goal. At the decision-making stage, only one of these methods is selected – a ***specific action plan***. By "pulling" all the accumulated experience from memory, motivational arousal is transformed into a ***specific goal*** that determines the way to achieve it. A specific goal is called "***higher motivation***" in the TFS.

Acceptor of action results. Motivational arousal also "extracts from memory" the entire sequence and hierarchy of results that must be achieved in order to complete the action plan. This sequence is called an ***acceptor of action results*** in the TFS. The acceptor of action results is the dominant need (Goal) of the body, transformed into the form of advanced brain stimulation, as if into a kind of ***complex receptor*** for future reinforcement, which is ***a criterion for achieving a specific goal***.

Reinforcements. ***The sanctioning stage***. If all the results of the acceptor of action results are achieved and the goal is achieved, then the last sanctioned stage occurs, in which the need is met and the specific action plan is stored in memory.

A formal model of TFS in terms of maximally specific causal connections (neurons) is given below.

11 Causality and Neurophysiology of Functional Systems Theory

Let us consider the TFS model that follows from neurophysiological data on neuronal excitations obtained in the TFS. Neurophysiologically, anticipation is realized by special collateral branches from the actions performed, which come to the "input" of the brain, converging with afferentation from input stimuli: "*We are talking about collateral branches of the pyramidal tract, which divert "copies" of those efferent messages that go to the pyramidal tract to many interstitial neurons ... Thus, the moment of decision-making and the beginning of the output of working efferent excitements (the beginning of actions – E.E.) from the brain is accompanied by the formation of an extensive complex of excitements consisting of afferent signs of a future result and a collateral "copy" of efferent excitements that went to the periphery along the pyramidal tract to the working organs*" Anokhin (1984).

In fact, this means the development of conditional (causal) connections between the implementation of actions (efferent excitements) and the subsequent perception of the results of actions represented by their afferent characteristics (see Fig. 3). When we perform actions, we immediately send a conditional signal via the collaterals that we are about to receive afferentation about the results of these

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

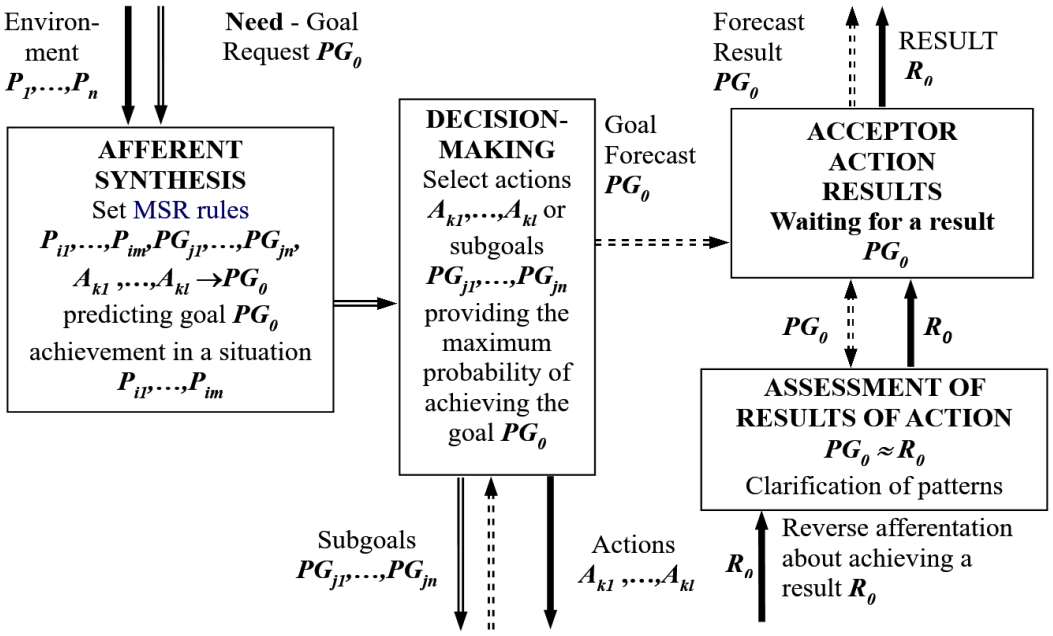


Figure 4: Scheme of the functional system

actions. This leads to the development of conditional (causal) relationships between actions and their results, reflecting the relationship of actions and results occurring in the external world. These are conditional connections made by the brain along ***the internal circuit*** (see Fig. 3), allow you to predict the results of actions taking place in the external world, even before the results themselves appear. When motivational arousal activates various sequences of actions to achieve the goal, then at the same time, the entire sequence and hierarchy of results that will be obtained in the process of achieving the goal is predicted along the "inner contour". When a decision is made on a specific action plan, the achievement of all intermediate results that make up the acceptor of action results is simultaneously anticipated along the "inner contour".

12 TFS MODEL IN TERMS OF MAXIMALLY SPECIFIC CAUSAL RELATIONSHIPS

We expand the previous scheme in terms of maximally specific causal relationships and corresponding excitations Fig. 4.

We will consider the need as a request to the functional system to achieve the goal indicated by the predicate PG_0. This request is sent to the afferent synthesis block. For functional systems that do not have functional subsystems, it extracts causal relationships of the form

$$P_{i1},\ldots,P_{im},A_{k1},\ldots,A_{kl} \Rightarrow PG_0$$

leading to goal achievement PG_0, where $P_{i1},\ldots,P_{im}$ is the environment properties required to achieve the goal, and $A_{k1},\ldots,A_{kl}$ is the sequence of actions leading to the goal. In this case the properties $P_{i1},\ldots,P_{im}$ must be conformed with the properties of the environment $P_1,\ldots,P_n$ that enter the afferent synthesis block.

For hierarchically functional systems that call functional subsystems, this query extracts more complex causal relationships from memory

$$P_{i1},\ldots,P_{im},PG_{j1},\ldots,PG_{jn},A_{k1},\ldots,A_{kl} \Rightarrow PG_0,$$

these include requests for achieving subgoals $PG_{j1},\ldots,PG_{jn}$. Then the extracted rules are sent to the decision-making block, where the goal achievement forecast is performed for each of the rules. The forecast of the rules, where only actions are performed, is based on the probability of the rule

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

itself. Forecasting based on rules with requests to subsystems is performed by sending these requests to these functional subsystems, making decisions in them and receiving probabilistic estimations of achieving these subgoals. The resulting forecast probability is calculated by multiplying the rule probability by the probability of achieving the subgoals.

After that, a decision is made for achieving the goal, and for this purpose, a rule is selected that has the maximum probability of predicting that the goal will be achieved. Then an action plan is formed that includes all actions included in the rule and all actions that are available in functional subsystems. Simultaneously with the action plan, the action results acceptor is formed, which includes the expectation of all predicted sub-results in functional subsystems and in the functional system itself. After that, the action plan is executed, and the expected results are compared with the results obtained.

If all the sub-results and the final result are achieved and coincide with the expected results, then the rule itself and all the rules of functional subsystems that were selected during the decision-making process are reinforced and their probability increases.

If the result is not achieved in a particular subsystem, the corresponding rule of this functional subsystem is penalized. After that, there is a tentative research reaction, which revises the decision to achieve the goal.

This model has been repeatedly refined and used for modeling animates Mukhortov et al. (2012); Demin & Vityaev (2014); Vityaev et al. (2020). The most revealing experiment was conducted with a nematode Demin & Vityaev (2014), when this model was embedded in an electronic model of the nematode, and after learning by "trial and error", this model learned its natural movement pattern used by a real nematode.

Thus, the cognitive group of functional systems has been successfully formalized.

13 Conclusion

Thus, we were able to show that by taking as a basis the principle that ***the brain detects all possible causal relationships in the external world and draws conclusions from them***, we can obtain mathematical models of the neuron, cellular ensembles in the form of probabilistic formal concepts and model of functional system.

These mathematical models are based on an accurate mathematical analysis of probabilistic causality and its formalization in the form of probabilistic maximally specific causal relationships, as well as on probabilistic generalization of formal concepts, which is a looping conclusion on maximally specific causal relationships.

Acknowledgments

The work was financially supported by the State Assignment of the Sobolev Institute of mathematics SB RAS for 2026-2029 "Theory of computability and logical aspects, logical calculus and their semantics" Supervisor: S.S. Goncharov. Project Number: FWNF-2026-0032.

References

The Nature of Classification. Relationships and Kinds in the Natural Sciences. Palgrave Macmillan, 2013.

K.V. Anokhin. The cognitome: Seeking the fundamental neuroscience of a theory of consciousness. *Neurosci Behav Physi*, 51, 2021.

P.K. Anokhin. *Biology and neurophysiology of the conditioned reflex and its role in adaptive behaviour*. Pergamon press, 1974.

P.K. Anokhin. Principled questions of the theory of functional systems. In *Philosophical aspects of the theory of functional systems*, pp. 49–106, Moscow, 1978.

P.K. Anokhin. *The general theory of functional systems*. Medicine, 1984.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1024 DOI: 10.66693/mathai.1024
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Rudolph Carnap. *Philosophical foundations of Physics*. Progress, 1971.

A.V. Demin and E.E. Vityaev. Learning in a virtual model of the c. elegans nematode for locomotion and chemotaxis, biologically inspired. *Biologically Inspired Cognitive Architectures*, 7, 2014.

Yurii Ershov and Klimentii Samokhvalov. On a new approach to the philosophy of mathematics. *Computer Systems*, 101:141–148, 1984.

Yurii Ershov and Klimentii Samokhvalov. *Sovremennaya filosofiya matematiki: nedomoganiya i lechenie*. Parallel Publ, 2007.

B. Ganter. *Formal Concept Analysis: Methods, and Applications in Computer Science*. TU Dresden, 2003.

B. Ganter and R. Wille. *Formal Concept Analysis. Mathematical Foundations*. TU Dresden, 2003.

J. Gibson. *Ecological approach to visual perception*. Progress, 1988.

D.O. Hebb. *The Organization of Behavior*. Wiley, 1949.

C.G. Hempel. Aspects of scientific explanation. In *Aspects of Scientific Explanation and other Essays*, Free Press, 1965.

C.G. Hempel. Maximal specificity and lawlikeness in probabilistic explanation. *Philosophy of Science*, 35, 1968.

B. Kovalerchuk and E. Vityaev. *Data Mining in Finance: Advances in Relational and Hybrid Methods*. Kluwer Acad. Pub., 2000.

J.S. Mill. *System of Logic. Ratiocinative and Inductive*. L., 1983.

V. Mukhortov, S. Khlebnikov, and E. Vityaev. Improved algorithm of semantic probabilistic inference in the 2-dimensional animate problem. *Neuroinformatics*, 6, 2012.

Masafumi Oizumi, Larissa Albantakis, and Giulio Tononi. From the phenomenology to the mechanisms of consciousness: Integrated information theory 3.0. *PLOS Computational Biology*, 10, 2014.

B. Rehder. Categorization as causal reasoning. *Cognitive Science*, 27, 2003.

Bob Rehder and Jay Martin. Towards a generative model of causal cycles. In *33rd Annual Meeting of the Cognitive Science Society 2011*, Boston, Massachusetts, 2011.

E Rosch. Principles of categorization. In *Cognition and Categorization*, Lawrence Erlbaum Associates, Publishers, 1978.

E.H. Rosch. Natural categories. *Cognitive Psychology*, 4, 1975.

E.S. Smirnov. Construction of a species from a taxonomic point of view. *Zool. Journal*, 17, 1938.

G. Tononi. Information integration: its relevance to brain function and consciousness. *Arch. Ital. Biol.*, 148, 2010.

G. Tononi. Integrated information theory of consciousness: an updated account. *Arch. Ital. Biol.*, 150, 2012.

E. Vityaev and S. Odintsov. How to predict consistently? In *Trends in Mathematics and Computational Intelligence, 796*, pp. 35–41, Springer, 2019.

E. Vityaev, A. Demin, and Y. Kolonin. Logical probabilistic biologically inspired cognitive architecture. In *Artificial General Intelligence*, pp. 12177, Springer, 2020.

E.E. Vityaev. *Extracting knowledge from data. Computer cognition. Modeling of cognitive processes*. NSU, 2006a.

Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

E.E. Vityaev. A formal model of neuron that provides consistent predictions. In *Biologically Inspired Cognitive Architectures 2012, Advances in Intelligent Systems and Computing, 196*, Springer, 2015.

E.E. Vityaev and V.V. Martinovich. Probabilistic formal concepts with negation. In *LNCS 8974*, Springer, 2015.

E.E. Vityaev and V.V. Martynovich. Formalization of "natural" classification and systematics through fixed points of predictions. *Sibirskie elektronnye matematicheskie izvestiya*, 12, 2015.

E.E. Vityaev, A.V. Demin, and Ponomarev D.K. Probabilistic generalization of formal concepts. *Programming*, 38, 2012.

Evgenii Vityaev. The logic of prediction. In *Proceedings of the 9th Asian Logic Conference*, pp. 263–276, World Scientific Publishers, 2006b.

Evgenii Vityaev. Mathematical probability model of cognitome and functional systems. In *Proceedings of the Conference "Interdisciplinary interaction of algebraic Biology, systems Theory and artificial Intelligence*, pp. 238–292, Moscow, 2021.

Evgenii Vityaev and Boris Kovalerchuk. Ontological data mining. In *Uncertainty Modeling: Dedicated to Professor Boris Kovalerchuk on his Anniversary*, pp. 277–292, Springer, 2017.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

DETECTING HALLUCINATIONS IN LLM RESPONSES USING TOKEN-LEVEL LOG-PROBABILITY SIGNALS

Vadim Eliseev, Aleksandra Maksimova

Laboratory of Intellegent Systems
Institute of Applied Mathematics and Mechanics, Donetsk, Russia
{eliseevv02,maximova.alexandra}@mail.ru

ABSTRACT

Large language models (LLMs) have proven themselves to be powerful tools for many natural language tasks — from being a high-quality text classifiers to acting as agents in complex retrieval-augmented generation (RAG) systems. However, from early beggining they suffer from a major limitation: hallucinations, i.e. confidently generating incorrect or misleading information that can also slightly correlate with the given task. This issue is critical in error-sensitive domains such as finance, medicine, and law, where even small inaccuracies can cause significant harm and detriment. In this study we address the early detection of hallucinating answers based on user input (prompt), answer by the LLM, and which is more important — token-level probabilty signals that can also be extracted from the LLM during its inference time. We constructed a dataset that combines textual information with sequences of token log-probabilities and their statistics (mean, min, variance, percentiles, etc.), labeled the answers whether they are hallucinations or not. We trained a lightweight classifier that outputs the probability that a given response is a hallucination. We evaluate the classifier and perform ablation studies to quantify the contribution of token-level signals versus text-only features. The intended use of the trained model is to be a standalone output guard agent in multi-agent system that rejects the answer of LLM-generator if its hallucination probability is above acceptance threshold and protects the users of it from having incorrect or misleading answer by making the whole system regenerate such answer or confirm that it cannot give the faithfull reply.

1 INTRODUCTION

Continuously advancing large language models (LLMs) such as LLaMA Touvron et al. (2023) family, newest versions of GPT Singh et al. (2025), Gemini Team et al. (2023) and others have demonstrated strong performance in a wide range of tasks beyond theoretical science and laboratory conditions. They are increasingly deployed in real-world applications, including customer support automation Landolsi et al. (2025), financial document analysis, legal assistance Xi et al. (2025), medical information systems Hassan et al. (2025), and retrieval-augmented enterprise workflows in both single and multi-agent approach Li et al. (2024).

Modern multi-agent systems are capable not only of answering textual queries using corporate knowledge data via classical retrieval-augmented generation (RAG) Lewis et al. (2020), but also of interacting with external tools and services through function calling and structured output generation, powered by ReAct Yao et al. (2022) approach. Instead of producing free-form text responses alone, contemporary LLM-based agents can generate machine-readable outputs (e.g., JSON schemas, API calls, database queries) that trigger downstream actions within larger software ecosystems. This shift transforms LLMs from purely generative text models into decision-making components embedded within executable workflows, enabling automation of complex business processes.

However, one of the most significant limitations of LLMs – **hallucinations** – persists even in state-of-the-art models.They hallucinating because their training and evaluation procedures reward guessing over acknowledging uncertainty Kalai et al. (2025). This issue becomes critical in real-world

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

applications, where LLMs playing role of a decision makers, especially when it comes about high-stakes domains such as finance, law or medicine. In this case, timely detection and mitigation of hallucinated responses can substantially reduce risks for both businesses and end users.

In this study we address the early detection of hallucinating LLM responses, focusing on applying proposed solution as an output guardrail agent within a multi-agent system. As the latency is a crucial constraint in real-world multi-agent deployments, we focus on building a lightweight hallucinations detection model that can be integrated into such systems without introducing substantial computational or latency overhead.

During the practical part of our study the following steps were performed:

1. Adopted an existing dataset containing user prompts and model-generated responses with additional features, initially lacked annotations for hallucinations;
2. Annotated the dataset with a binary label (0 or 1) indicating whether a given model response constitutes a hallucination;
3. Analyzed the labeled dataset, extracted the suitable for classification features from token log-probabilities;
4. Trainied the basic versions of classical ML algorithms on labeled dataset in order to find the best baseline classifier;
5. Performed hyperparameter tuning of the best classifier from previous step in order to improve model performance.

These steps will be described in the following sections, the source code of the project is open-source and available at:

`https://github.com/EliseevVadim/advanced-hallucinations-detection`

2 Related Work

Hallucinations in LLMs have been a subject of extensive research from both theoretical and applied perspectives. Early work highlighted that LLMs may produce fluent yet factually incorrect or unsupported content, often with high confidence Ji et al. (2023). Subsequent studies analyzed the underlying causes of hallucinations, including exposure bias, limitations of next-token prediction objectives, and misalignment between training objectives and factual correctness Kalai et al. (2025). The hallucinations taxonomy provided later at Huang et al. (2025) shows the different types of hallucinations combined with their key reasons that can help engineers mitigate them more precise.

In practical side, hallucination mitigation has been approached from several complementary directions. One of the most common strategies is retrieval-augmented generation (RAG), which grounds model outputs in external knowledge sources to improve factual consistency Lewis et al. (2020). However, RAG systems are still hallucinating, so this cannot be used as a ”silver bullet” as it is. Another line of work focuses on post-generation verification, where either rule-based systems or secondary LLMs act as judges to assess factual correctness and consistency Kadavath et al. (2022). LLM-based hallucination detection pipelines are also discussed at Anaokar et al. (2025). Other frameworks such as HuDEx integrate hallucination detection with explainability, enabling models not only to flag problematic outputs but also provide interpretable error explanations Lee et al. (2025). However these approaches might not be perfect in real-world scenarios as it invokes another LLM that should be at least as good or better than our generator which causing computational and latency overhead. Also LLM-as-a-judge pipelines considering only textual inputs, but in practice we can obtain also some numerical internal information about model reply, such as token probabilities (which we are focusing in current study), internal representations from LLMs layers, etc.

Lightweight neural models like HaluNet Tong et al. (2025) leverage multi-granular uncertainty modeling, fusing token probabilities with semantic embedding information to deliver efficient one-pass detection suitable for real-time applications. Other approaches address both detection and calibration by capturing hallucination signals within output and semantic spaces, such as HaDeMiF Zhou et al. (2025), which uses compact decision trees and neural networks to calibrate and mitigate hallucinations in models outputs during inference. While these approaches have demonstrated

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

improvements in factual robustness, they still introduce additional computational overhead, require external knowledge sources, or depend on auxiliary models, which may limit their suitability for latency-sensitive multi-agent systems.

3 METHOD

To address the incompleteness of LLM-based hallucinations detectors (as they process textual inputs only) and complexity of systems, based on neural networks, combined with LLMs we propose a lightweight hallucination detection framework based on classical machine learning algorithms. This enables us to take into account both textual features and internal generation information (e.g., token-level probabilities and model hidden states).

In order to do this we utilized the publicly available dataset Massaron (2024) which contains responses generated by Mistral 7B Instruct Jiang et al. (2023) to prompts from the Open Orca dataset Lian et al. (2023). In addition to textual information, the dataset includes structured generation metadata such as the number of tokens in the response (according to Mistral 7B tokenizer), token-level probabilities, and the generation finish reason. The dataset in initial form does not contain hallucination labels. Therefore, an explicit annotation procedure was required prior to supervised training. The preprocessing and labeling pipeline is described in detail in Section 4.

Following dataset annotation, we constructed a collection of baseline classifiers using classical machine learning algorithms initialized with default hyperparameters. This stage was designed to identify the most suitable model for the task independently of hyperparameter optimization. For textual representation, we employed sparse vectorization techniques which performing effective in combination with linear classifiers for text classification tasks Joachims (1998). Sparse representations are often preferable for classical ML pipelines, as dense neural embeddings may not provide consistent advantages outside deep architectures and can be suboptimal for linear models Reimers and Gurevych (2019). We evaluated different combinations of textual encoders and classification algorithms to determine the most effective detection pipeline. The procedure is described in Section 5.1.

After selecting the best classification pipeline the hyperparameter optimization was applied. We identified the most influential parameters of the chosen classifier and defined appropriate search ranges. A search procedure was then applied to determine the optimal hyperparameter configuration according to the selected evaluation metric. Details of the hyperparameter tuning stage and final configuration are provided in Section 5.2.

4 DATA PREPARATION AND ANALYSIS

This section outlines the preprocessing, exploratory data analysis, and labeling procedures applied to the dataset making it suitabile for supervised training of classification pipelines.

4.1 RAW DATA PREPROCESSING

As the source of training data we utilized the publicly available dataset Massaron (2024). It is based on Open Orca dataset Lian et al. (2023) and contains responses generated by Mistral 7B Instruct Jiang et al. (2023) to its prompts. The dataset contains following attributes: the original prompt, the reference answer, the model-generated answer, decoding parameters (temperature and top-p), the number of completion tokens (i.e., the length of the generated response in tokens), token log-probabilities of each generated token, and the generation finish reason (indicating whether the generation terminated naturally or was truncated due to the maximum output length constraint).

After an initial exploratory analysis of the dataset, we observed that each prompt is associated with **21** generations, resulting in a total of **392,091** records. Such a large number of responses per prompt is redundant and will cause computational overhead during the labeling process, so we decided to reduce its size. To achieve this while preserving informative variability, we introduced a scalar *risk score*– a proxy feature designed to estimate the likelihood that a given generation may contain hallucinated content. This feature was used only for dataset size reduction and was removed in all subsequent training stages.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

The *risk score* was computed using the token log-probabilities of each generated response. Let $\{\ell_i\}_{i=1}^n$ denote the log-probabilities of the generated tokens, where n is the number of tokens in the response.

We define

$$\mu = \frac{1}{n}\sum_{i=1}^{n}\ell_i \tag{1}$$

as the mean log-probability,

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\ell_i - \mu)^2} \tag{2}$$

as the standard deviation of log-probabilities,

$$f = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}(\ell_i < \tau) \tag{3}$$

as the fraction of low-confidence tokens, where τ denotes a predefined confidence threshold and $\mathbf{1}(\cdot)$ is the indicator function.

The risk function is then defined as

$$\mathrm{Risk}(\{\ell_i\}) = \frac{-\alpha\mu + \beta\,\sigma + \gamma\,f}{\log(n+1)} \tag{4}$$

where $\alpha, \beta, \gamma \geq 0$ are tunable coefficients that control the contribution of each component to the overall risk score.

In the current implementation, the coefficients were set to $\alpha = 1$, $\beta = 0.5$, $\gamma = 0.3$ and $\tau = -5.0$.

The components of the risk function can be interpreted as follows:

- $-\mu$ increases the risk when the model's average confidence is low (the negative sign ensures that lower mean log-probability results in higher risk);
- σ increases the risk when the token-level predictions exhibit high variability;
- f amplifies the contribution of tokens with particularly low confidence;
- division by $\log(n+1)$ normalizes the risk with respect to response length.

Additionally, we observed that all generations were produced with a fixed top-p parameter of 0.95, while the temperature took only two values: **0.0** or **0.8**. Based on this observation, we computed the risk score for each generation and performed group-wise selection within each prompt according to the following criteria:

- Row with temperature = 0.0 as it is most deterministic one;
- Row with lowest risk score in the group (most likely not a hallucination);
- Row with biggest risk score in the group (most likely a hallucination);
- Row with closest to median risk score in the group (most interesting examples for classification).

Applying this risk-based filtering strategy reduced the dataset size to **69,451** records, which is approximately six times smaller than the original dataset.

Finally, we removed records in which the generated answer did not contain any numeric characters or alphabetic symbols (Latin or Cyrillic), as such outputs were considered non-informative for the classification task. This additional filtering step eliminated another **1,056** records.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

4.2 DATA LABELING

After reducing the dataset size, the remaining samples were labeled whether each generated response contains hallucinated content. We adopted a semi-automatic labeling strategy based on two complementary quantitative metrics.

For each record, we computed the following scores:

- $h \in [0, 1]$ – the `hallucination_score`, obtained from the Google Gemma-3 12B model Kamath et al. (????) as a judge via using the prompt shown in Figure 4.2;
- $s \in [0, 1]$ – the `similarity_score`, representing the cosine similarity Salton et al. (1975) between the generated answer and the reference answer. Vector representations of both answers were obtained using *nomic-embed-text-v1.5* model Nussbaum et al. (2024)

SYSTEM
You are a world-class state-of-the-art assistant for determining whether a Model Answer is a hallucination / contains hallucinations or not based on a Prompt or an Expected Answer.
Your task is to estimate the likelihood that the Model Answer contains hallucinations with respect to the Prompt and the Expected Answer.
Definition of Hallucination for this task:
A hallucination occurs if the Model Answer contains:
1. Factual Inaccuracy: Statements that contradict the Expected Answer.
2. Internal Nonsense: The answer is self-contradictory, nonsensical, misleading, erroneous or otherwise semantically incoherent.
3. Unsupported Fabrication: Information (facts, details, conclusions) that is not contained in, cannot be inferred from, or contradicts the provided Prompt.
You must output a single floating-point number in the range [0.0, 1.0], representing the continuous likelihood of hallucination.
Important requirements:
-The score MUST be a continuous value, not a discrete category.
- Do NOT snap, round, or map your score to fixed anchor values.
- Avoid simple fractions such as 0.25, 0.5, 0.75 unless they are genuinely exact.
- Prefer nuanced values (e.g., 0.13, 0.42, 0.67) when appropriate.
- Use up to 5 decimal places.
Guidance for scoring (DO NOT copy numbers literally; use them as intuition):
- Answers that exactly match the Expected Answer should be 0.0.
- Answers that are mostly correct, with minor discrepancies, should be slightly higher.
- Answers that are partially supported, with some unclear or misleading facts, should be in the middle.
- Answers that are mostly unsupported or contain many misleading or incorrect facts should be high.
- Answers that are empty, only line breaks, completely nonsensical, misleading, erroneous or unrelated to the Prompt should be at the maximum.
Rules:
- Rephrasing or paraphrasing of the Expected Answer is NOT hallucination.
- Stylistic or formatting differences are NOT hallucination.
- Output MUST be valid JSON.
- Output MUST contain ONLY the JSON object, no explanations.
Analysis Steps (Consider in order):
1. Factual Accuracy: How closely does it match the Expected Answer's facts and sense?
2. Groundedness: Is the Model Answer supported by and relevant to the Prompt?
3. Fabrication: Does it introduce information absent from the Prompt?
4. Coherence: Is the answer self-consistent and logically structured?

INPUT
Prompt: {prompt}
Expected Answer: {expected_answer}
Model Answer: {answer}

Figure 1: Prompt template used to obtain hallucination scores from the judge LLM.

Let $\alpha, \beta \geq 0$ be weighting coefficients such that

$$\alpha + \beta = 1. \tag{5}$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

We define the aggregated target score as a normalized weighted sum:

$$\text{target_score} = \alpha h + \beta(1 - s). \tag{6}$$

Here, α controls the contribution of the hallucination score, while β determines the influence of semantic dissimilarity. The term $(1 - s)$ is used to align both components in the same direction: since s measures cosine similarity, $(1 - s)$ corresponds to cosine distance Sahoo and Maiti (2025), ensuring that higher values consistently indicate a higher likelihood of hallucination. The constraint $\alpha + \beta = 1$ guarantees a normalized linear combination.

The final binary target variable is defined using a threshold τ:

$$y = \begin{cases} 1, & \text{if target_score} \geq \tau \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

In the experimental configuration, we set

$$\tau = 0.5. \tag{8}$$

Thus, responses with $\text{target_score} \geq 0.5$ are assigned to the positive class (hallucinated), while the remaining samples are labeled as non-hallucinated. Adjusting α and β allows controlling the relative sensitivity of the labeling procedure to hallucination severity versus semantic deviation from the reference answer.

After obtaining hallucination scores from Gemma and computing the cosine similarity values, we performed a grid search over α and β in the interval $[0, 1]$ with step of 0.1. For each combination, we computed the positive class fraction. The corresponding heatmap is presented in Figure 2.

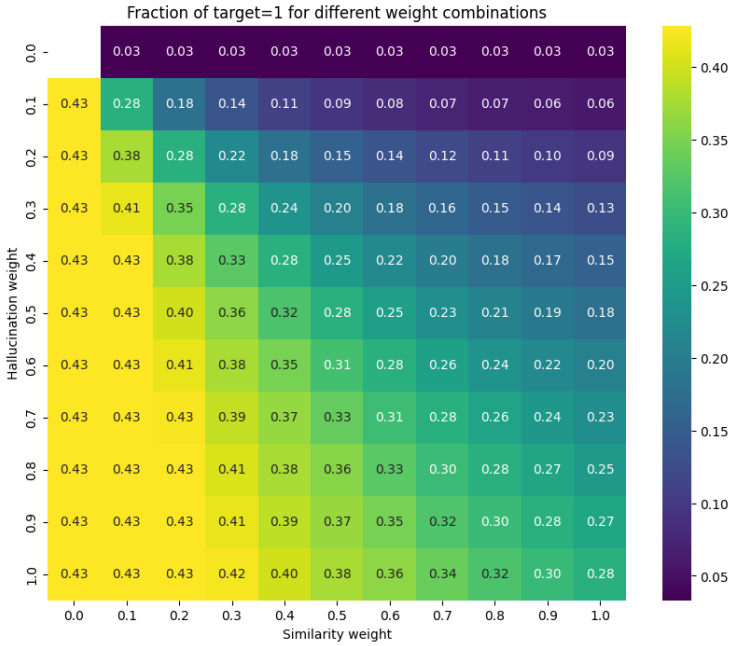


Figure 2: Positive class fraction for different weight combinations

As a result, we selected the coefficient values $\alpha = 0.6$ and $\beta = 0.4$, which yield approximately 35% of samples assigned to the positive class and 65% to the negative class. This distribution reflects a slight class imbalance avoiding extreme disproportionality preserving realistic asymmetry in the data. The chosen weighting is also supported by the intuition that the LLM-as-a-judge score should have a stronger influence than the semantic distance metric, while not fully determining the decision.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

4.3 FINAL FEATURES EXTRACTION

As mentioned above, one of the core features expected to improve the performance of the classification pipeline are token log-probabilities of each generated response. However, classical machine learning algorithms require input in the form of fixed-length numerical feature vectors and cannot directly consume variable-length lists of floating-point numbers without first transforming them into a uniform representation suitable for model input Le and Mikolov (2014). It is necessary to derive a set of summary statistics that capture the essential characteristics of the log-probability distributions and can be represented as a set of features for downstream classification.

Let $\{\ell_i\}_{i=1}^n$ denote the token-level log-probabilities of a generated response.

As previously defined in Equations 1, 2, and 3, we compute the mean log-probability μ, the standard deviation σ, and the fraction of low-confidence tokens f as the separated features too. They characterize the overall confidence level, variance of token-level certainty, and the prevalence of highly uncertain tokens, respectively.

In addition to these features, we extract the following complementary ones.

Median log-probability

The median provides a robust estimate of central tendency that is less sensitive to extreme values than the mean:

$$\text{median}(\ell_1, \ldots, \ell_n) \tag{9}$$

Minimum log-probability:

$$\ell_{\min} = \min_{1 \le i \le n} \ell_i. \tag{10}$$

This feature captures the most uncertain token in the sequence and reflects the strongest local drop in model confidence during generation.

Maximum log-probability

$$\ell_{\max} = \max_{1 \le i \le n} \ell_i. \tag{11}$$

Although less directly related to hallucination risk, this statistic provides complementary information about peak confidence levels.

Normalized entropy of token probabilities

To measure the global variance of confidence across tokens, we convert log-probabilities into normalized weights:

$$w_i = \frac{e^{\ell_i}}{\sum_{j=1}^n e^{\ell_j}}. \tag{12}$$

Then we compute the Shannon entropy Shannon (1948):

$$H = -\sum_{i=1}^{n} w_i \log w_i. \tag{13}$$

To ensure comparability across sequences of different lengths, we normalize the entropy:

$$H_{\text{norm}} = \frac{H}{\log n}. \tag{14}$$

By construction,

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

$$0 \leq H_{\text{norm}} \leq 1. \tag{15}$$

Values close to 0 indicate that confidence is concentrated in a small subset of tokens, whereas values close to 1 correspond to a more uniformly distributed confidence pattern.

The correlation matrix between numeric features is shown in Figure 3.

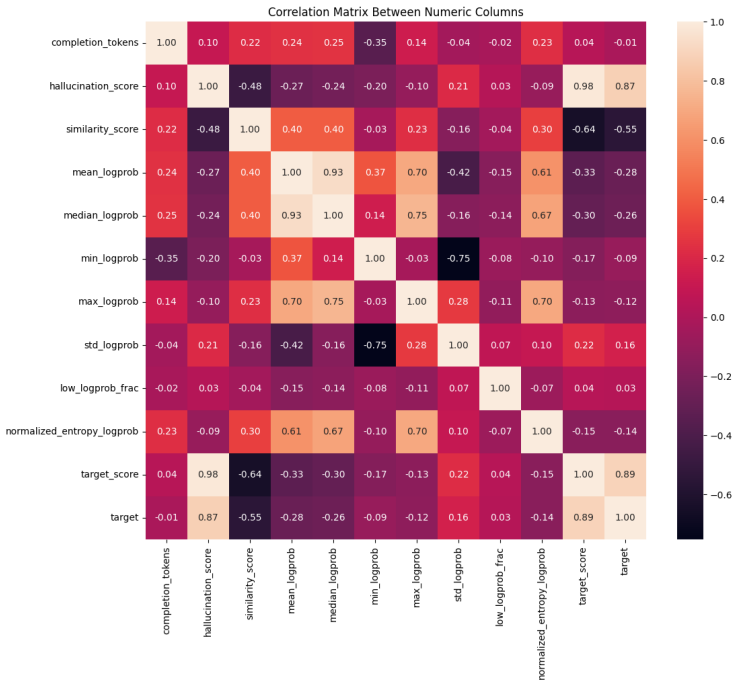


Figure 3: Correlation matrix between numberic features in the dataset

The correlation matrix does not reveal strong linear dependencies between the features, indicating the absence of pronounced multicollinearity. A moderate correlation can be observed between the target variable and the statistical descriptors of token log-probabilities. This suggests that log-probability-based features may carry informative signals relevant for classfication and are likely to contribute meaningfully to the performance of the trained models.

5 Classifiers Training

This section outlines the process of selecting the best classification pipeline, which consists of both a textual feature vectorization component and a classification model. We evaluate different combinations of vectorization techniques and classifiers to determine the configuration that achieves the best performance on the target task and then tuning hyperparameters of best pipeline.

5.1 Best Classification Pipeline Selection

After successfully building the dataset we can finally train our classifier. As classification models not able to process raw textual data, we need to vectorize *prompt* and *answer* columns. As we work on building a lightweight model based on classical machine learning algorithm, we need to make a sparse embeddings rather than dense ones Joachims (1998). For this we applied two sparse vectorizing approaches – Bag-of-Words (Count Vectorizer) and TF-IDF and compared their performance with different classification algorithms.

After constructing the dataset, we proceed to the training of classification models. Since classical machine learning algorithms cannot directly operate on raw textual data, the *prompt* and *answer*

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

fields must first be transformed into numerical representations. As we work on building a lightweight classification pipeline based on traditional machine learning methods, we employ sparse vector representations rather than dense embeddings Joachims (1998). We applied two widely used sparse vectorization approaches: the Bag-of-Words model (Count Vectorizer) Harris (1954) and Term Frequency–Inverse Document Frequency (TF–IDF) weighting Salton and Buckley (1988). We evaluate both representations in combination with several classification algorithms to determine the most effective pipeline configuration.

As the classification models we compare the following algorithms: Dummy Classifier (predicts the objects as a most frequent class from training data), Logistic Regression Cox (1958) (as we solving the task of binary classification), Decision Tree classifier Breiman et al. (2017), Random Forest Breiman (2001) two gradient boosting implementations – LightGBM Ke et al. (2017) and CatBoost Prokhorenkova et al. (2018).

We evaluate all combinations of vectorizers and classifiers using an out-of-fold validation scheme Stone (1974) based on 5-fold cross-validation. To preserve the class distribution across folds, we employ stratified k-fold splitting Kohavi et al. (1995), which maintains approximately the same proportion of target classes in each fold.

Model performance is assessed using several standard metrics for binary classification. The primary evaluation metric is the Area Under the Receiver Operating Characteristic Curve (ROC AUC) Hanley and McNeil (1982), as it measures the model's ability to rank positive instances higher than negative ones independently of a fixed decision threshold. This is especially relevant for our task, where the classification threshold may be adjusted at inference time. Also we report F1-score van Rijsbergen (1979), precision, recall, accuracy, and the training time of each pipeline to compare computational efficiency. All experiments were conducted on a MacBook Pro M4 Max (36 GB RAM, 1 TB SSD).

We also applied stratified k-fold splitting in order to ensure that each fold gonna have the same target distribution. For classifiers quality evaluation we used the following metrics, applicable for task of binary classification: ROC AUC score (the most important metric for us as we want be able change positive class threshold during the inference), F1-score, precision, recall, accuracy and training time of each pipeline (all experiments were conducted on MacBook Pro M4 Max 36GB/1TB).

All classifications models were initialized with their default parameters as we aim to find most suitable one and then tune its hyperparameters. The performance metrics of the classification pipelines based on the Bag-of-Words (BoW) vectorizer are presented in Figure 4.

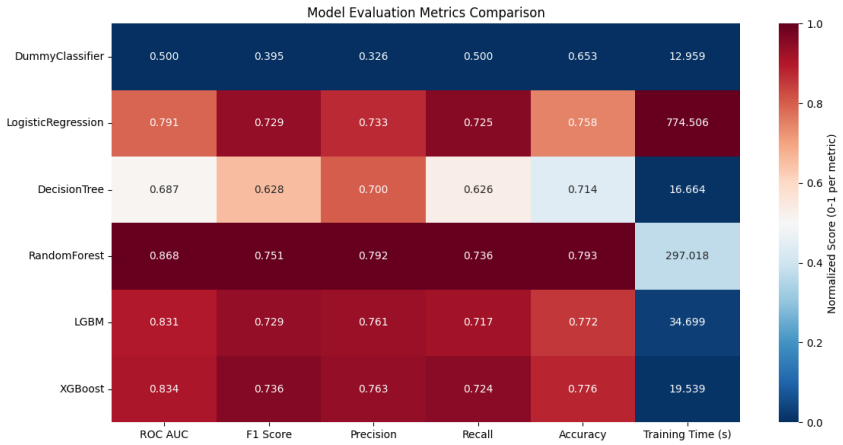


Figure 4: Metrics of pipelines based on BoW vectorization

The best performance was shown by tree-based classifiers: Random Forest and gradient boosting implementations. In Figure 5 the performance metrics of the classification pipelines based on the TF-IDF vectorizer are presented.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

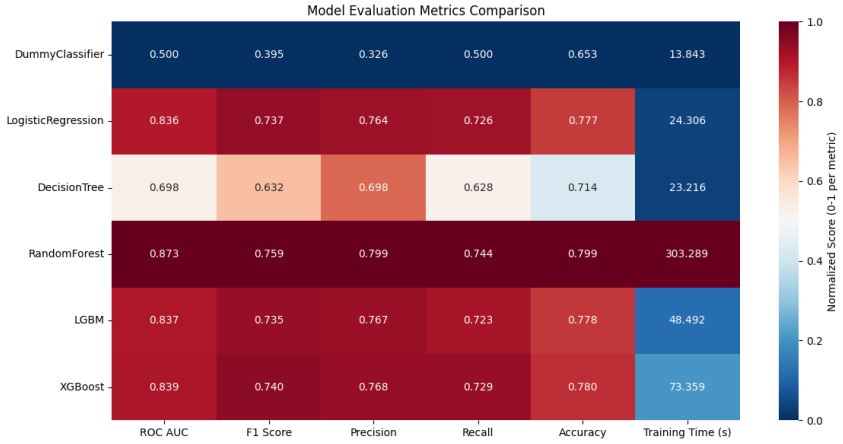


Figure 5: Metrics of pipelines based on TF-IDF vectorization

As we see, the best performance was shown by tree-based estimators too. A noticeable improvement can be observed for Logistic Regression. This indicates that the TF-IDF representation provides a more informative and discriminative vectorization of textual features compared to the alternative sparse encoding.

5.2 Best classifier tuning

On the previous step we found, that the best performative pipeline for current task is the combination of TF-IDF vectorizer and Random Forest. In this section we describe its tuning process.

In the previous stage, we identified the combination of TF-IDF vectorization and Random Forest as the most effective pipeline for the current task. In this section, we describe the hyperparameter tuning procedure for this model.

We tune the following hyperparameters:

Number of trees ($n_{\text{estimators}}$). This parameter controls the size of the ensemble. We consider the following values:

$$n_{\text{estimators}} \in \{20, 50, 100, 300\}.$$

Increasing the number of trees generally reduces variance and improves stability, at the cost of higher computational time.

Maximum tree depth (max_depth). This parameter limits the depth of each decision tree:

$$\text{max_depth} \in \{\text{None}, 5, 10, 20\}.$$

Restricting tree depth reduces model complexity and helps prevent overfitting, while allowing unlimited depth (None) enables fully grown trees.

The splitting criterion that determines how node impurity is measured during tree construction. We evaluate three commonly used criteria:

- Gini impurity,
- entropy (information gain),
- log-loss.

We used Halving Search as the best parameters search algorithm and ROC AUC as the optimizing metric because this is the target one for our task.

For hyperparameter optimization, we employ the Successive Halving search strategy Li et al. (2018). As the optimization objective, we use ROC AUC, since it serves as the primary evaluation metric for our task.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Successive Halving is a resource-efficient hyperparameter search algorithm that iteratively allocates increasing computational resources to the most promising candidate configurations while eliminating poorly performing ones at early stages. This process continues until a small set of high-performing configurations remains. Compared to exhaustive grid search, Successive Halving significantly reduces computational cost while maintaining competitive selection quality.

After completing the search the following hyperparameters were selected:

- $criterion$ – Gini,
- max_depth – None,
- $n_{estimators}$ – 300.

The confusion matrix and ROC curve of the best pipeline is shown in Figure 6.

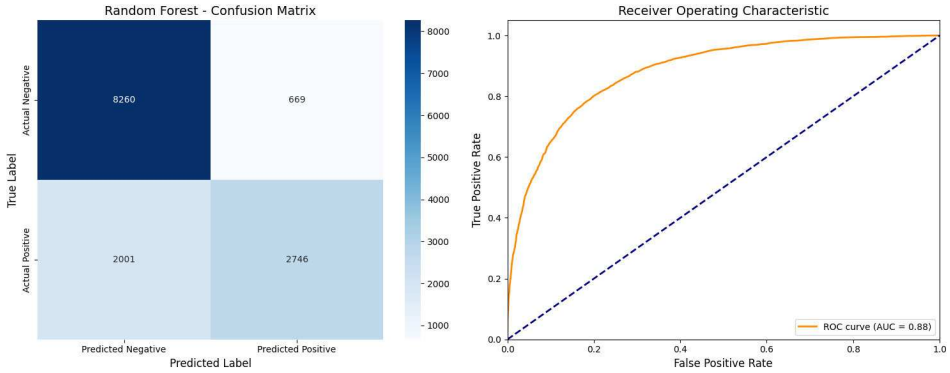


Figure 6: Confusion matrix and ROC curve of the best classification pipeline

Parameters tuning increased the ROC AUC of the pipeline to **0.88** which is a decent metric for such lightweight classifier.

6 Future work

After successfully developing a lightweight hallucination detector, we plan to integrate it as an output guardrail within the multi-agent architecture of the Donetsk State University intelligent assistant.

Two primary integration strategies are considered. The first approach is to deploy the trained classifier directly as a post-generation filtering module, flagging potentially unreliable responses before they are delivered to the end user. The second one involves extending the training dataset by incorporating generations produced by LLaMA 3.1 8B Instruct[1] and re-running the full training and evaluation pipeline described in this study.

7 Conclusion

In this paper, we presented a process of labeling LLM-generated responses with respect to hallucination presence, given a prompt and an expected answer. Based on this labeled dataset, we evaluated multiple lightweight classification pipelines combining sparse textual vectorization methods (Bag-of-Words and TF-IDF) with classical machine learning algorithms. The novelty of our study lies in using not only textual features for determining hallucinations in the responses, but also by incorporating token log-probabilities of the response (presented by extracted scalar features from them).

The best-performing pipeline – TF-IDF combined with a tuned Random Forest classifier achieved a ROC AUC score of **0.88**, demonstrating that lightweight models can provide strong performance in hallucination detection without relying on large neural architectures. The feature importances of the best classification pipeline are shown in Figure 7.

[1] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1025 DOI: 10.66693/mathai.1025
Special Issue: MathAI 2026 Selected Papers
Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

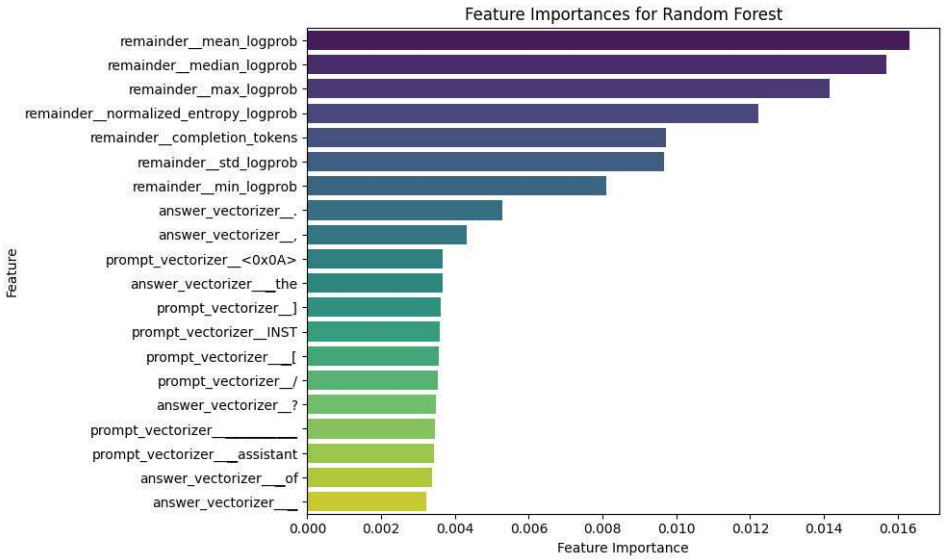


Figure 7: Top-20 features with their importances of the best classification pipeline

Analysis of feature importances reveals that the features derived from token log-probabilities significantly outperform pure textual ones, confirming our hypothesis that generation-time uncertainty signals play a crucial role in hallucination detection. These findings highlight the effectiveness of combining sparse textual representations with probabilistic confidence-based features in building efficient and interpretable hallucination detection systems.

ACKNOWLEDGMENTS

The study was carried out with the financial support of the Ministry of Education and Science of the Russian Federation within the framework of the state task on the topic ”Development and improvement of intelligent classification and forecasting methods for pattern recognition and modeling of information processes” FREM-2024-0001 (Registration number 1023111000141-9-1.2.1)

REFERENCES

Spandan Anaokar, Shrey Ganatra, Swapnil Bhattacharyya, Harshvivek Kashid, Shruthi N Nair, Reshma Sekhar, Siddharth Manohar, Rahul Hemrajani, and Pushpak Bhattacharyya, Anaokar, Spandan and Ganatra, Shrey and Bhattacharyya, Swapnil and Kashid, Harshvivek and Nair, Shruthi N and Sekhar, Reshma and Manohar, Siddharth and Hemrajani, Rahul and Bhattacharyya, Pushpak (2025). Halludetect: Detecting, mitigating, and benchmarking hallucinations in conversational systems in the legal domain. pp., 1822–1847.

Leo Breiman, Breiman, Leo (2001). Random forests. *Machine learning*, 45(1):5–32.

(2017). *Classification and regression trees*. Chapman and Hall/CRC.

David R Cox, Cox, David R (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 20(2):215–232.

James A Hanley and Barbara J McNeil, Hanley, James A and McNeil, Barbara J (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Zellig S Harris, Harris, Zellig S (1954). Distributional structure. *Word*, 10(2-3):146–162.

Tasnimul Hassan, Md Faisal Karim, Haziq Jeelani, Elham Behnam, Robert Green, and Fayeq Jeelani Syed, Hassan, Tasnimul and Karim, Md Faisal and Jeelani, Haziq and Behnam, Elham and Green,

Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Robert and Syed, Fayeq Jeelani (2025). Optimizing medical question-answering systems: A comparative study of fine-tuned and zero-shot large language models with rag framework. *arXiv preprint arXiv:2512.05863*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al., Huang, Lei and Yu, Weijiang and Ma, Weitao and Zhong, Weihong and Feng, Zhangyin and Wang, Haotian and Chen, Qianglong and Peng, Weihua and Feng, Xiaocheng and Qin, Bing and others (2025). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung, Ji, Ziwei and Lee, Nayeon and Frieske, Rita and Yu, Tiezheng and Su, Dan and Xu, Yan and Ishii, Etsuko and Bang, Ye Jin and Madotto, Andrea and Fung, Pascale (2023). Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed, Albert Q. Jiang and Alexandre Sablayrolles and Arthur Mensch and Chris Bamford and Devendra Singh Chaplot and Diego de las Casas and Florian Bressand and Gianna Lengyel and Guillaume Lample and Lucile Saulnier and Lélio Renard Lavaud and Marie-Anne Lachaux and Pierre Stock and Teven Le Scao and Thibaut Lavril and Thomas Wang and Timothée Lacroix and William El Sayed (2023). Mistral 7b. URL `https://arxiv.org/abs/2310.06825`.

Thorsten Joachims, Joachims, Thorsten (1998). Text categorization with support vector machines: Learning with many relevant features. pp., 137–142, Springer.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al., Kadavath, Saurav and Conerly, Tom and Askell, Amanda and Henighan, Tom and Drain, Dawn and Perez, Ethan and Schiefer, Nicholas and Hatfield-Dodds, Zac and DasSarma, Nova and Tran-Johnson, Eli and others (2022). Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Adam Tauman Kalai, Ofir Nachum, Santosh S Vempala, and Edwin Zhang, Kalai, Adam Tauman and Nachum, Ofir and Vempala, Santosh S and Zhang, Edwin (2025). Why language models hallucinate. *arXiv preprint arXiv:2509.04664*.

A Kamath, J Ferret, S Pathak, N Vieillard, R Merhej, S Perrin, T Matejovicova, A Ramé, M Rivière, L Rouillard, et al., Kamath, A and Ferret, J and Pathak, S and Vieillard, N and Merhej, R and Perrin, S and Matejovicova, T and Ramé, A and Rivière, M and Rouillard, L and others (????). Gemma 3 technical report. corr, abs/2503.19786, 2025. doi: 10.48550. *arXiv preprint ARXIV.2503.19786*.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu, Ke, Guolin and Meng, Qi and Finley, Thomas and Wang, Taifeng and Chen, Wei and Ma, Weidong and Ye, Qiwei and Liu, Tie-Yan (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Ron Kohavi et al., Kohavi, Ron and others (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. volume and number, pp., 1137–1145, Montreal, Canada.

Hamza Landolsi, Kais Letaief, Nizar Taghouti, and Ines Abdeljaoued-Tej, Landolsi, Hamza and Letaief, Kais and Taghouti, Nizar and Abdeljaoued-Tej, Ines (2025). Caprag: A large language model solution for customer service and automatic reporting using vector and graph retrieval-augmented generation. *arXiv preprint arXiv:2501.13993*.

Quoc Le and Tomas Mikolov, Le, Quoc and Mikolov, Tomas (2014). Distributed representations of sentences and documents. pp., 1188–1196, PMLR.

Sujeong Lee, Hayoung Lee, Seongsoo Heo, and Wonik Choi, Lee, Sujeong and Lee, Hayoung and Heo, Seongsoo and Choi, Wonik (2025). Hudex: Integrating hallucination detection and explainability for enhancing the reliability of llm responses. *arXiv preprint arXiv:2502.08109*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al., Lewis, Patrick and Perez, Ethan and Piktus, Aleksandra and Petroni, Fabio and Karpukhin, Vladimir and Goyal, Naman and Küttler, Heinrich and Lewis, Mike and Yih, Wen-tau and Rocktäschel, Tim and others (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Jiaxing Li, Chi Xu, Lianchen Jia, Feng Wang, Cong Zhang, and Jiangchuan Liu, Li, Jiaxing and Xu, Chi and Jia, Lianchen and Wang, Feng and Zhang, Cong and Liu, Jiangchuan (2024). Eaco-rag: Towards distributed tiered llm deployment using edge-assisted and collaborative rag with adaptive knowledge update. *arXiv preprint arXiv:2410.20299*.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar, Li, Lisha and Jamieson, Kevin and DeSalvo, Giulia and Rostamizadeh, Afshin and Talwalkar, Ameet (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of machine learning research*, 18(185):1–52.

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium", Wing Lian and Bleys Goodson and Eugene Pentland and Austin Cook and Chanvichet Vong and "Teknium" (2023). Openorca: An open dataset of gpt augmented flan reasoning traces. `https://https://huggingface.co/datasets/Open-Orca/OpenOrca`.

Luca Massaron, Luca Massaron (2024). Genai-hallucinations dataset. `https://www.kaggle.com/datasets/lucamassaron/genai-hallucinations`. Dataset of model responses (from Open-Orca prompts) using a Mistral 7B Instruct LLM labeled for hallucination research.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar, Zach Nussbaum and John X. Morris and Brandon Duderstadt and Andriy Mulyar (2024). Nomic embed: Training a reproducible long context text embedder.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin, Prokhorenkova, Liudmila and Gusev, Gleb and Vorobev, Aleksandr and Dorogush, Anna Veronika and Gulin, Andrey (2018). Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Nils Reimers and Iryna Gurevych, Reimers, Nils and Gurevych, Iryna (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. pp., 3982–3992.

Satyajeet Sahoo and Jhareswar Maiti, Sahoo, Satyajeet and Maiti, Jhareswar (2025). Variance-adjusted cosine distance as similarity metric. *arXiv preprint arXiv:2502.02233*.

Gerard Salton and Christopher Buckley, Salton, Gerard and Buckley, Christopher (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Gerard Salton, Anita Wong, and Chung-Shu Yang, Salton, Gerard and Wong, Anita and Yang, Chung-Shu (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Claude Elwood Shannon, Shannon, Claude Elwood (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al., Singh, Aaditya and Fry, Adam and Perelman, Adam and Tart, Adam and Ganesh, Adi and El-Kishky, Ahmed and McLaughlin, Aidan and Low, Aiden and Ostrow, AJ and Ananthram, Akhila and others (2025). Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Received: 05 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Mervyn Stone, Stone, Mervyn (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al., Team, Gemini and Anil, Rohan and Borgeaud, Sebastian and Alayrac, Jean-Baptiste and Yu, Jiahui and Soricut, Radu and Schalkwyk, Johan and Dai, Andrew M and Hauth, Anja and Millican, Katie and others (2023). Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Chaodong Tong, Qi Zhang, Jiayang Gao, Lei Jiang, Yanbing Liu, and Nannan Sun, Tong, Chaodong and Zhang, Qi and Gao, Jiayang and Jiang, Lei and Liu, Yanbing and Sun, Nannan (2025). Halunet: Multi-granular uncertainty modeling for efficient hallucination detection in llm question answering. *arXiv preprint arXiv:2512.24562*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample, Hugo Touvron and Thibaut Lavril and Gautier Izacard and Xavier Martinet and Marie-Anne Lachaux and Timothée Lacroix and Baptiste Rozière and Naman Goyal and Eric Hambro and Faisal Azhar and Aurelien Rodriguez and Armand Joulin and Edouard Grave and Guillaume Lample (2023). Llama: Open and efficient foundation language models. URL `https://arxiv.org/abs/2302.13971`.

C. J. van Rijsbergen, van Rijsbergen, C. J. (1979). Information retrieval. *Butterworths*.

Yueqing Xi, Yifan Bai, Huasen Luo, Weiliang Wen, Hui Liu, and Haoliang Li, Xi, Yueqing and Bai, Yifan and Luo, Huasen and Wen, Weiliang and Liu, Hui and Li, Haoliang (2025). Hybrid retrieval-augmented generation agent for trustworthy legal question answering in judicial forensics. *arXiv preprint arXiv:2511.01668*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao, Yao, Shunyu and Zhao, Jeffrey and Yu, Dian and Du, Nan and Shafran, Izhak and Narasimhan, Karthik R and Cao, Yuan (2022). React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Xiaoling Zhou, Mingjie Zhang, Zhemg Lee, Wei Ye, and Shikun Zhang, Zhou, Xiaoling and Zhang, Mingjie and Lee, Zhemg and Ye, Wei and Zhang, Shikun (2025). Hademif: Hallucination detection and mitigation in large language models. In *The Thirteenth International Conference on Learning Representations*.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1026 DOI: 10.66693/mathai.1026
Special Issue: MathAI 2026 Selected Papers
Received: 06 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

ESTIMATING IMPORTANCE OF HIGHLY CORRELATED FEATURES USING MATRIX FACTORIZATION

Alexander S. Minkin[1],*

[1] *Keldysh Institute of Applied Mathematics*
Moscow,125047, Russia
amink@mail.ru

ABSTRACT

Hyperspectral images contain a large volume of source data that exhibits high correlations along neighboring spectral bands. This makes it necessary to select the most informative features among correlated groups of features to effectively solve various machine learning problems. A method of feature importance evaluation for hyperspectral image data is proposed. This method combines iterative training of Decision Tree classifiers based on spectral features with matrix factorization to overcome sparsity. Decision trees provide intrinsic feature selection mechanism but only a small number of features are usually taken into account by the CART algorithm for training a single decision tree classifier instance. Furthermore when features are highly correlated (e.g., Pearson $\rho > 0.8$), tree-based methods like Random Forest or XGBoost arbitrarily assign importance to one feature while suppressing others, as they redundantly capture the same signal. To overcome this problem, an additional balancing term was incorporated into the optimization function used to obtain the matrix factorization.

The considered method of feature importance evaluation is compared with such model-specific tree-based methods as vanilla Gini impurity decrease and more complicated Boruta algorithm. Classification accuracy is tested using a Random Forest classifier on significant features. Selecting features with higher importance scores yields models boasting greater training accuracy.

Keywords: feature selection, tree-based methods, matrix factorization, machine learning, hyperspectral images

1 INTRODUCTION

Hyperspectral images (HSI) usually acquired from satellites capture light spectra in numerous narrow wavelength ranges. This approach boosts their informational content compared to conventional RGB images. HSI processing applications, including cloud detection, monitoring, agriculture and environmental protection (Borzov and Uzilov, 2016), are challenged by the high dimensionality inherent in large volumes of HSI data. These applications typically rely on machine learning (ML) methods, which can be adversely affected by the curse of dimensionality. A family of techniques designed to overcome the curse of dimentionality are commonly addressed through a set of methods known as dimentionality reduction. In this paper we consider the method for such reduction by selecting a relevant subset of features from both the original and derived sets, without applying any transformations. Furthermore, the high correlations between neighboring spectral bands lead to significant data redundancy. To address this issue and improve the efficiency of subsequent analysis, a special method of feature selection is employed here to identify and retain only the most informative spectral bands and derived features.

The existing methods of feature selection assess both feature properties and target variable relationships, covering forward selection and backward elimination methods, exhaustive search, and ML-based techniques (Guyon and Elisseeff, 2003). Feature importance evaluation can be done directly by analyzing the pair relations between feature and target variable. This approach is generally

*Please send copies to the alternative email address amink@mail.ru

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1026 DOI: 10.66693/mathai.1026
Special Issue: MathAI 2026 Selected Papers
Received: 06 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

called filter methods. Correlations between adjacent spectral bands gives folks not only to the idea of using dimentionality reduction algorithms for image compression but also to select relevant features (Myasnikov, 2017) from which PCA is the most popular (Zimichev et al., 2014). This approach along with direct analysis of the correlation matrix does not require additional labeling. Such methods as mutual information scores and ANOVA F-value are also very popular filter methods that takes into account labels but doesn't rely on training algorithms. Though these methods are computationally efficient they ignore feature interdependence which can lead to the selection of redundant features that are highly correlated with each other (Zhu et al., 2007). That is why, to select features more effectively, information from the results of HSI classification, in combination with the existing labeling of the source data, is needed. ML models are widely used to evaluate the importance of features, especially in cases of feature correlations.

Model-specific methods for feature selection integrate the feature selection process directly into the model training phase. This approach offers good balance of performance and computational cost (Saito et al., 2018). There are wrapper and embedded model-specific methods. Wrapper methods identify the optimal subset of features by evaluating their different combinations using a specific predictive model. Models such as the recursive feature elimination (RFE) select features iteratively, maximizing the performance of the classification or regression models. Although effective in finding optimal subsets, these methods can be computationally demanding (Zubair et al., 2024). Embedded methods integrate feature selection directly into the model training process itself, using mechanisms like regularization penalties to perform selection simultaneously with parameter estimation. Examples of embedded methods include Lasso (Least Absolute Shrinkage and Selection Operator) and Ridge regression (Fira et al., 2025), neural networks with learnable drop layer (JimÉnez-Navarro et al., 2024), sparse principal component analysis (Seghouane et al., 2019), tree-based methods (Tuv et al., 2009) and so on.

Decision trees provide an intrinsic feature selection mechanism (Breiman et al., 1984), effectively handling non-linear data and correlated features Kohavi and John (1997), and offering inherent explainability Mishra (2022). Decision tree-based models like Random Forests and Gradient Boosting provide feature importance scores based on how much a feature contributes to reducing impurity (e.g., Gini impurity decrease) or variance at each split in the decision trees. More complicated Boruta method determine feature relevance by comparing the importance of an original feature with the importance of permuted counterparts known as shadow features and functions as a wrapper algorithm built around the Random Forest classifier (Kursa and Rudnicki, 2010). As the method tests multiple combinations of original and shadow features to assess feature importance, the process is quite time-consuming.

In general model specific methods give better results than filter methods but their significant drawback is that the selected set of features is inherently tied to the specific ML algorithm being used (Islam et al., 2022). Stochastic nature of some ML models can lead to reproducibility issues, with variations in feature importance rankings across different model training runs (Vos et al., 2024).

Model-agnostic methods offer greater flexibility and are widely used for interpreting complex "black-box" models. Model-agnostic feature selection methods can be applied to any machine learning model, as they are independent of the model's internal workings (Khan et al., 2025). These methods assess the relevance of features based on their intrinsic properties and their relationship with the target variable, without considering a specific predictive model. So they usually require a model to be trained first, but the method itself doesn't care which trainable model is used. Examples of model-agnostic techniques include methods based on eXplainable Artificial Intelligence (XAI) like Permutation Feature Importance (PFI) (Flora et al., 2024) and SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017). These techniques offer flexibility and can be used to compare the importance of features across different models but computationally intensive and still affected by feature correlations (Liang et al., 2024) especially PFI (Salih, 2025).

The more sophisticated method of feature importance evaluation is used the more computing time it requires. The more features are taken into account, the more their importance is affected by multicollinearity. In scenarios with a high degree of feature correlation, a Decision Tree classifier offers an efficient way to assess how various feature combinations influence training outcomes, as it operates quickly and is robust to multicollinearity. However for correlated features tree-based methods can assign disproportionate importance to one feature at the expense of others because they redundantly model the same predictive signal. In this case Gini impurity decrease is generally considered

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1026 DOI: 10.66693/mathai.1026
Special Issue: MathAI 2026 Selected Papers
Received: 06 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

misleading for evaluating feature importance and needs correction. The method of such correction is the topic of the present paper. Multiple evaluations of feature importance across different subsets result in slightly varying and sparse importance values. This paper proposes a method for approximating feature importance values extracted from a sparse matrix, which aggregates results from multiple evaluations on different feature subsets. We suggest the algorithm of feature importance correction by means of matrix factorization obtained via gradient descent process.

2 Problem statement and method of solution

While high-dimensional HSI data offers rich information content, it also presents significant challenges that can degrade the performance of ML models. The aim of this work is to develop an algorithm that enhances model effectiveness by selecting an optimal subset of spectral bands and derived features, thereby improving the prediction accuracy for dense cloud classification.

The iterative training of ML models with different subset of features is used here to approximate target variable. This is done by feature elimination algorithm (Minkin, 2026) by training different instances of Decision Tree classifiers to get multiple values of feature importance. The importance of spectral bands and derived features is determined by assessing how their selection influences the decrease in Gini impurity within Decision Tree classifiers. This approach allows us to rank features according to their contribution to model performance. By evaluating feature subsets during training, we construct a sparse matrix of feature importance scores, with rows corresponding to features and columns to individual trained decision trees. This approach yields the following observations:

1. Minimal subset of features is considered because of correlations between them;
2. Training ML models using as many features as possible is computationally expensive;
3. As a result of multiple training of decision trees, the feature importance matrix is sparse.
4. For correlated features A and B when the tree looks for the best split, it might choose feature A simply because of a tiny, random fluctuation in the data that makes its Gini score marginally better than feature B's at that specific node. Feature A gets credited with the entire impurity decrease for that split. Feature B is ignored at that node because the information it provides is redundant. So the selection becomes arbitrary. In some trees, A is chosen; in others, B is chosen. This leads to unstable feature importance scores. Hence, multiple assessments of feature importance helps reveal their true predictive value.

Logistic regression as linear model are strongly affected by feature correlations (Pourhoseingholi et al., 2008). Decision trees are advantageous here, but training with large feature subsets is still computationally expensive. The resulting importance matrix is sparse, which gives rise to the problem of zero value reconstruction. Such reconstruction is needed to overcome possible feature importance underestimation for correlated features when using Gini impurity decrease as a metric of such evaluation in case of classification problem. When feature importance is approximated from a subset of entries in the sparse matrix, highly correlated features are expected to yield similar importance scores. That process should incorporate additional term during optimization to yield matrix factorization as the result.

So we reformulated the problem by the following way. Feature importance are modeled as the following matrix factorization inspired by recommendation systems key concepts and algorithms from Koren et al. (2009):

$$\hat{T} = PQ^T, \tag{1}$$

where $\hat{T}$ $(n_u \times n_i)$ is the predicted importance corresponding to trained ML model instance u and feature i, P $(n_u \times n_f)$ and Q $(n_i \times n_f)$ are latent factors that capture hidden preferences for features and train instances, respectively. The challenge to the matrix factorization problem is to find P and Q^T. Basically, such an algorithm is going to be used to find latent factors that represent intrinsic feature attributes in a lower dimension (the number n_f of latent factors is chosen beforehand). A learning approach is therefore developed to converge the factorization results close to the observed importance as much as possible, while ensuring all importance values remain nonnegative. Additionally we introduce the feature bias matrix μ as a correction of (1):

$$\hat{T}_{u,i} = \hat{\mu}_{u,i} + \hat{p}_u \hat{q}_i^T. \tag{2}$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1026 DOI: 10.66693/mathai.1026
Special Issue: MathAI 2026 Selected Papers
Received: 06 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

The feature bias matrix μ supposed to capture the tendency of features to have importance higher (or lower) than the average. So μ measures a trained model tendency to systematically overestimate or underestimate feature importance relative to the average across all features and trained ML model instances. Matrices P, Q and μ can be obtained through a regularized optimization procedure:

$$\sum_u \sum_i \left((T_{u,i} - \hat{T}_{u,i})^2 + \lambda(\hat{\mu}_{u,i}^2 + ||\hat{p}_u||^2 + ||\hat{q}_i||^2) \right). \tag{3}$$

Feature importance for correlated features is a key challenge in machine learning interpretability, as standard methods often split or misattribute importance unpredictably. SHAP values become unstable with high multicollinearity (e.g., Pearson $\rho > 0.8$), where one feature might capture all credit due to model fitting order or randomness. PFI can amplify this by perturbing one feature while others compensate via correlation. Correlated features convey redundant information, causing tree-based methods like Gini importance in Random Forest or XGBoost to distribute scores roughly equally among them, masking true group contributions. To overcome randomness in feature importance distribution via tree-based methods the matrix factorization approach is used here with the following additional contribution to the optimization function:

$$\sum_u \sum_i \left((T_{u,i} - \hat{T}_{u,i})^2 + \lambda(\hat{\mu}_{u,i}^2 + ||\hat{p}_u||^2 + ||\hat{q}_i||^2) \right) + E_{u,i,j}, \tag{4}$$

$$E_{u,i,j} = \sum_u \sum_i \sum_j r_{i,j}^2 T_{u,j} (T_{u,j} - \hat{T}_{u,i})^2, \tag{5}$$

where $r_{i,j}$ is Pearson correlation between features i and j. The term $E_{u,i,j}$ in equation (5) penalizes discrepancies in importance scores among highly correlated features, thereby encouraging consistent importance estimates. The penalty term scales positively with both $T_{i,j}$ and the magnitude of the importance discrepancy among correlated features.The derived term takes larger values for features with low importance and smaller values for highly important features.

Stochastic Gradient Descent used here to solve the problem (2) of matrix factorization is an optimization algorithm in which the model parameters (in this case, the bias $\mu_{u,i}$ and factor vectors $\hat{p}_u$ and $\hat{q}_i$) are repeatedly updated by adding the negative of gradients calculated with respect to the function (4) being optimized. The algorithm essentially performs the following steps for a given number of iterations:

$$\begin{aligned}
\hat{\mu}_{u,i} &\leftarrow \hat{\mu}_{u,i} + \gamma \left(\Delta_{u,i} - \lambda \hat{\mu}_{u,i} \right) \\
\hat{p}_u &\leftarrow \hat{p}_u + \gamma \left(\Delta_{u,i} \cdot \hat{q}_i - \lambda \hat{p}_u \right) \\
\hat{q}_i &\leftarrow \hat{q}_i + \gamma \left(\Delta u, i \cdot \hat{p}_u - \lambda \hat{q}_i \right) \\
\Delta_{u,i} &= \delta_{u,i} + \sum_{j \neq i} r_{i,j}^2 T_{u,j} \delta_{u,i,j}
\end{aligned} \tag{6}$$

where γ is the learning rate $\delta_{u,i} = T_{u,i} - \hat{T}_{u,i} = T_{u,i} - (\mu_{u,i} + p_u q_i^T)$ is the error made by the model for the pair (u, i) and $\delta_{u,i,j} = T_{u,j} - \hat{T}_{u,i} = T_{u,j} - (\mu_{u,i} + p_u q_i^T)$ denotes the pairwise distance between the importance score of feature j and the approximated importance score of feature i.

3 Experiments

3.1 Setup

A dedicated set of labeled images of the HYPERION sensor with a spatial resolution of 30 m and a spectral resolution of 10 m in the spectral range of 400–2500 nm was used. After converting the raw radiance data into reflectance values and eliminating zero channels as well as those corresponding to strong light absorption in water vapor, the selected HSI were organised into a training set. The features of this set comprised reflectance values from spectral channels 8–224, combined with derived characteristics and other indices calculated on a pixel-by-pixel basis. Fig. 1 shows the mean spectral reflectance distribution for cloud (green line) and non-cloud (red line) pixels, suggesting that a classification algorithm could be developed to separate these classes based on the spectral characteristics of pixels. In this study, in addition to reflectance values, normalized indices such as NDVI (Huang et al., 2021), NDSI (Jin et al., 2022) and NDWI (Gao, 1996) were also used but they are not shown in Fig. 1.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1026 DOI: 10.66693/mathai.1026
Special Issue: MathAI 2026 Selected Papers
Received: 06 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

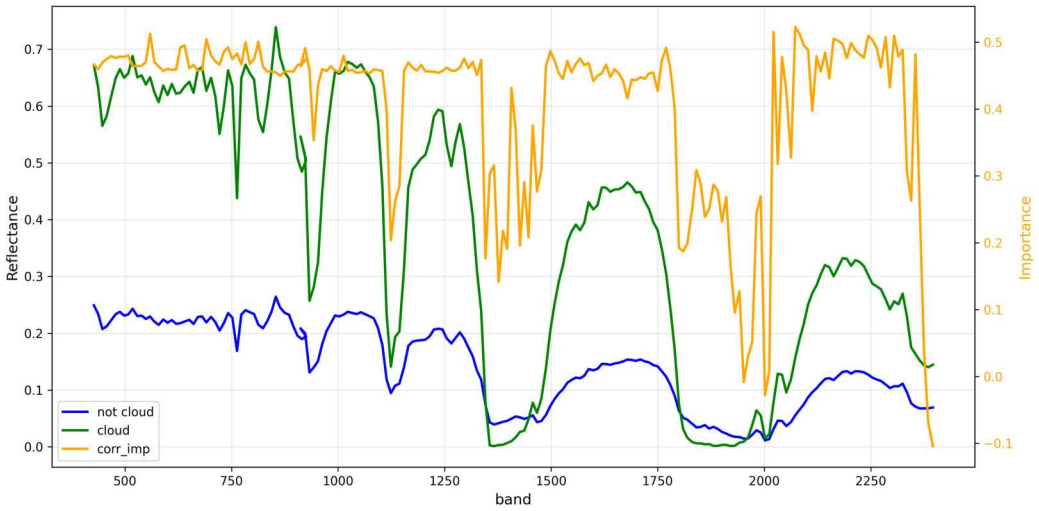


Figure 1: Mean spectral reflectance curves: cloud pixels (green), non-cloud pixels (blue), and correlation feature importance scores (red)

3.2 Results

The input data for the matrix factorization algorithm is feature importance statistics evaluated by training Decision Tree classifier for different values of hyperparameters. For Decision Tree classifier the decrease of Gini impurity is used to assess the feature importance. By considering the accuracy associated with different feature sets, we can calculate the correlation between feature importance and training accuracy and sort features according to this value of correlation. Let's refer to this as correlation feature importance. Fig. 1 shows the effect of assigning greater importance to features with larger differences in spectral reflectance between cloud and non-cloud pixels.

The most important features include NDWI index and the limited set of features from the NIR and the lower SWIR wavelength range. In general there is little to no similarity between the feature importance rankings obtained via the Boruta algorithm and those from the present study, with the exception of the consistently high importance assigned to the NDWI feature and the lower SWIR wavelength range. This can be explained by the use of shadow features in feature analysis via the Boruta algorithm not used here. Fig. 2 presents the random forest model's accuracy as a function

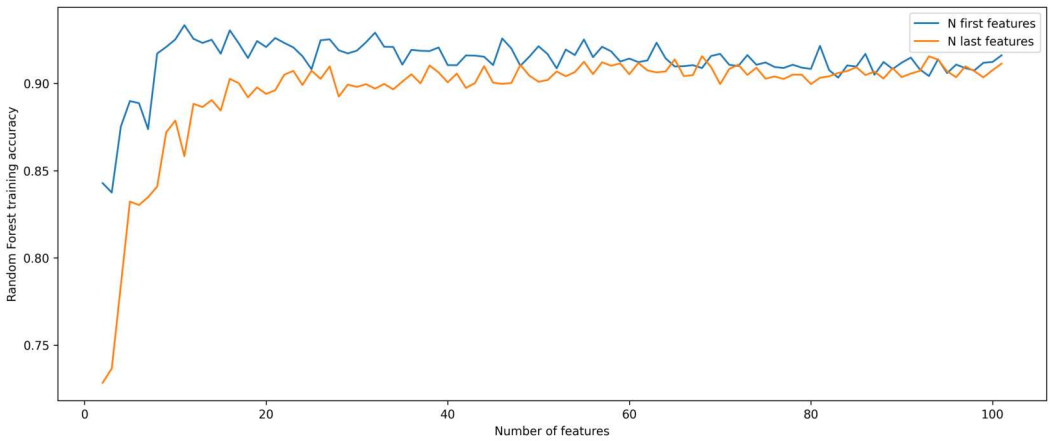


Figure 2: Random Forest training for the first and the last features by their correlation with training accuracy of different Decision Tree classifiers

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1026 DOI: 10.66693/mathai.1026
Special Issue: MathAI 2026 Selected Papers
Received: 06 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

of the number of selected features. The blue curve illustrates the performance of Random Forest classifier when features are sequentially added following a descending ranking by correlation-based importance (top or first N features). Conversely, the orange curve shows the outcome of adding features in ascending order of importance (bottom or last N features from the same pre-sorted list). Fig. 2 shows that choosing features with higher importance scores (starting from the top of the list in descending order of importance), we obtain models with higher training accuracy.

4 Conclusion

The significant number of channels in HSI, combined with feature multicollinearity, leads to challenges in selecting machine learning models, reducing their accuracy and interpretability. To address this issue for dense cloud classification, decision trees are employed with a selection of a limited number of significant features based on the iterative exclusion algorithm. The proposed method optimizes feature selection by leveraging the correlation between feature importance scores and training accuracy. The most relevant features include NDWI index, limited number of NIR bands and the lower part of SWIR spectrum range bands. Decision Tree model used to assess feature importance effectively handles correlated features avoiding redundancy and can be combined with forward feature selection or backward elimination to achieve robust statistics. The final set of features selected after applying matrix factorization to overcome sparsity with balancing feature importance scores and taking into account the correlation between feature importance and model accuracy. This can be used to construct a classifier for recognizing dense clouds based on their spectral characteristics. Such classifier can be considered a baseline for more complex cloud classification models.

The ambiguity in feature importance rankings produced by traditional methods – such as PFI, SHAP, tree-based methods, and linear model coefficients – underscores the need for a more empirically grounded approach. The proposed strategy of iterative feature elimination with feature importance approximation via matrix factorization offers an alternative solution to this challenge. This approach not only reveals which features are truly critical for maintaining performance but also accounts for feature interactions and helps identify their optimal subset. One of the way to evaluate feature importance is to use Gini impurity decrease in combination with special algorithm for matrix factorization. This study introduces a feature importance evaluation method that combines Gini impurity decrease with a specialized matrix factorization algorithm to approximate the importance of correlated features, refining the final scores using classification accuracy feedback. This was done by rankings derived from the correlation between approximated feature importance scores and classification accuracy to obtain the list of features sorted by their importance.

References

S.M. Borzov and S.B. Uzilov, Borzov, S.M. and Uzilov, S.B. (2016). *J. Comput. Technol.*, 21(1):40 – 48.

(1984). *Classification and Regression Trees*. Chapman and Hall/CRC.

Monica Fira, Liviu Goras, and Hariton-Nicolae Costin, Fira, Monica and Goras, Liviu and Costin, Hariton-Nicolae (2025). Evaluating sparse feature selection methods: A theoretical and empirical perspective. 15(7).

Montgomery L. Flora, Corey K. Potvin, Amy McGovern, and Shawn Handler, Montgomery L. Flora and Corey K. Potvin and Amy McGovern and Shawn Handler (2024). A machine learning explainability tutorial for atmospheric sciences. *Artificial Intelligence for the Earth Systems*, 3(1):e230018. URL `https://journals.ametsoc.org/view/journals/aies/3/1/AIES-D-23-0018.1.xml`.

B.-C. Gao, Gao, B.-C. (1996). Ndwi—a normalized difference water index for remote sensing of vegetation liquid water from space. *Remote Sensing of Environment*, 58(3):257–266.

Isabelle Guyon and André Elisseeff, Guyon, Isabelle and Elisseeff, André (2003). An introduction of variable and feature selection. *J. Machine Learning Research Special Issue on Variable and Feature Selection*, 3:1157 – 1182.

Received: 06 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Sha Huang, Lina Tang, Joseph P. Hupy, Yang Wang, and Guofan Shao, Huang, Sha and Tang, Lina and Hupy, Joseph P. and Wang, Yang and Shao, Guofan (2021). A commentary review on the use of normalized difference vegetation index (ndvi) in the era of popular remote sensing. *Journal of Forestry Research*, 32(1):1–6.

Md Rashedul Islam, Aklima Akter Lima, Sujoy Chandra Das, M. F. Mridha, Akibur Rahman Prodeep, and Yutaka Watanobe, Islam, Md Rashedul and Lima, Aklima Akter and Das, Sujoy Chandra and Mridha, M. F. and Prodeep, Akibur Rahman and Watanobe, Yutaka (2022). A comprehensive survey on the process, methods, evaluation, and challenges of feature selection. *IEEE Access*, 10:99595–99632.

M J JimÉnez-Navarro, M MartÍnez-Ballesteros, I S Brito, F MartÍnez-Álvarez, and G Asencio-CortÉs, JimÉnez-Navarro, M J and MartÍnez-Ballesteros, M and Brito, I S and MartÍnez-Álvarez, F and Asencio-CortÉs, G (2024). Embedded feature selection for neural networks via learnable drop layer. *Logic Journal of the IGPL*, 33(5):jzae062.

Donghyun Jin, Kyeong-Sang Lee, Sungwon Choi, Noh-Hun Seong, Daeseong Jung, Suyoung Sim, Jongho Woo, Uujin Jeon, Yugyeong Byeon, and Kyung-Soo Han, Donghyun Jin and Kyeong-Sang Lee and Sungwon Choi and Noh-Hun Seong and Daeseong Jung and Suyoung Sim and Jongho Woo and Uujin Jeon and Yugyeong Byeon and Kyung-Soo Han (2022). An improvement of snow/cloud discrimination from machine learning using geostationary satellite data. *International Journal of Digital Earth*, 15(1):2355–2375.

Adam Khan, Asad Ali, Jahangir Khan, Fasee Ullah, and Muhammad Faheem, Khan, Adam and Ali, Asad and Khan, Jahangir and Ullah, Fasee and Faheem, Muhammad (2025). Exploring consistent feature selection for software fault prediction: An xai-based model-agnostic approach. *IEEE Access*, 13:75493–75524.

Ron Kohavi and George H. John, Ron Kohavi and George H. John (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324. URL `https://www.sciencedirect.com/science/article/pii/S000437029700043X`. Relevance.

Yehuda Koren, Robert Bell, and Chris Volinsky, Koren, Yehuda and Bell, Robert and Volinsky, Chris (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Miron B. Kursa and Witold R. Rudnicki, Kursa, Miron B. and Rudnicki, Witold R. (2010). Feature selection with the boruta package. *Journal of Statistical Software*, 36(11):1–13.

Annie Liang, Thomas Jemielita, Andy Liaw, Vladimir Svetnik, Lingkang Huang, Richard Baumgartner, and Jason M. Klusowski, Annie Liang and Thomas Jemielita and Andy Liaw and Vladimir Svetnik and Lingkang Huang and Richard Baumgartner and Jason M. Klusowski (2024). Challenges in variable importance ranking under correlation.

Scott Lundberg and Su-In Lee, Lundberg, Scott and Lee, Su-In (2017). A unified approach to interpreting model predictions.

A.S. Minkin, Minkin, A.S. (2026). *Atmospheric and Oceanic Optics*, 39(3):286–295.

(2022). *Practical explainable AI using python: Artificial Intelligence Model Explanations Using Python-Based Libraries, Extensions, and Frameworks*. New York: Apress.

E.V. Myasnikov, Myasnikov, E.V. (2017). Hyperspectral image segmentation using dimensionality reduction and classical segmentation approaches. *Comput. Opt.*, 41(4):564–572.

Mohamad Amin Pourhoseingholi, Yadolah Mehrabi, Hamid Alavi-Majd, and Parvin Yavari, Pourhoseingholi, Mohamad Amin and Mehrabi, Yadolah and Alavi-Majd, Hamid and Yavari, Parvin (2008). Using latent variables in logistic regression to reduce multicollinearity, a case-control example: breast cancer risk factors. *Italian Journal of Public Health*, 5(1).

Shota Saito, Shinichi Shirakawa, and Youhei Akimoto, Saito, Shota and Shirakawa, Shinichi and Akimoto, Youhei (2018). Embedded feature selection using probabilistic model-based optimization. p., 1922–1925. Association for Computing Machinery.

Received: 06 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Ahmed M. Salih, Ahmed M. Salih (2025). Re-visiting explainable ai evaluation metrics to identify the most informative features.

Abd-Krim Seghouane, Navid Shokouhi, and Inge Koch, Seghouane, Abd-Krim and Shokouhi, Navid and Koch, Inge (2019). Sparse principal component analysis with preserved sparsity pattern. *IEEE Transactions on Image Processing*, 28(7):3274–3285.

Eugene Tuv, Alexander Borisov, George Runger, and Kari Torkkola, Tuv, Eugene and Borisov, Alexander and Runger, George and Torkkola, Kari (2009). Feature selection with ensembles, artificial variables, and redundancy elimination. *Journal of Machine Learning Research*, 10:1341–1366.

Gideon Vos, Liza van Eijk, Zoltan Sarnyai, and Mostafa Rahimi Azghadi, Gideon Vos and Liza van Eijk and Zoltan Sarnyai and Mostafa Rahimi Azghadi (2024). Stabilizing machine learning for reproducible and explainable results: A novel validation approach to subject-specific insights.

Zexuan Zhu, Yew-Soon Ong, and Manoranjan Dash, Zhu, Zexuan and Ong, Yew-Soon and Dash, Manoranjan (2007). Wrapper–filter feature selection algorithm using a memetic framework. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(1):70–76.

E.A. Zimichev, N.L. Kazanskiy, and P.G. Serafimovich, Zimichev, E.A. and Kazanskiy, N.L. and Serafimovich, P.G. (2014). Spectral-spatial classification with k-means++ particional clustering. *Comput. Opt.*, 38(2):281–286.

Iqbal Muhammad Zubair, Yung-Seop Lee, and Byunghoon Kim, Zubair, Iqbal Muhammad and Lee, Yung-Seop and Kim, Byunghoon (2024). A new permutation-based method for ranking and selecting group features in multiclass classification. *Applied Sciences*, 14(8).

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1030 DOI: 10.66693/mathai.1030
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

OPERATOR LEARNING FOR HIGH-DIMENSIONAL SYMPLASTIC GROWTH DYNAMICS WITH STOCHASTIC CELL DIVISION

Ulyana Zubairova
Department of Information Technologies,
Novosibirsk State University,
Novosibirsk, Russia
u.zubairova@g.nsu.ru

ABSTRACT

We study operator learning for a nonlinear dynamical system describing symplastic plant leaf growth with multiple interacting cell files and stochastic cell division. The biomechanical model consists of coupled ordinary differential equations governing visible cell lengths, relaxed wall lengths, isosmotic lengths, and shared wall fragments. For N cell files with M cells per file, the state dimension scales as $D(N, M) \sim 3NM + K(N, M)$, where K denotes the number of shared fragments. While the number of state variables grows linearly in N, fragment-based mechanical coupling induces a rapidly increasing interaction structure, leading to dense Jacobians and growing computational cost of numerical integration. In the multi-file regime, repeated simulation becomes computationally prohibitive for parameter exploration and inverse calibration. We formalize the simulator as a nonlinear operator $\mathcal{F} : \Theta \subset \mathbb{R}^p \to \mathbb{R}^B$ mapping mechanical parameters to the longitudinal cell length profile. We train multilayer perceptron (MLP) surrogates to approximate $\mathcal{F}$ using simulator-generated data. The learned surrogate replaces repeated ODE integration and enables fast prediction of spatial growth profiles. We evaluate generalization performance on held-out parameter configurations and demonstrate efficient parameter calibration to experimental profiles. We further analyze structural properties of the parameter-to-profile map, including local regularity and an intrinsic stochastic noise floor induced by random cell division. Our results show that neural operator approximation provides a scalable framework for accelerating analysis and inverse modeling of coupled high-dimensional biological growth dynamics.

1 INTRODUCTION

Understanding how plant leaves grow under mechanical and osmotic constraints is central to developmental biology and crop modeling. Cell-based models of symplastic growth represent the tissue as a network of mechanically coupled cells sharing wall fragments. In particular, Zubairova et al. (2016) proposed a quasi-one-dimensional biomechanical model that explicitly expresses osmotic and turgor pressures and reproduces experimentally observed longitudinal cell length distributions in the wheat leaf epidermis.

From a dynamical systems perspective, this framework defines a high-dimensional hybrid stochastic ODE. Continuous biomechanical evolution of cell lengths and wall properties is coupled with discrete cell division events, while stochastic division factors introduce variability in the resulting profiles. For N cell files with M cells per file, the state dimension scales as $D(N, M) \sim 3NM + K(N, M)$, where $K(N, M)$ is the number of shared fragments. Fragment-mediated mechanical coupling produces dense Jacobians, and the computational cost of numerical integration increases with the number of coupled cell files. Repeated simulation therefore becomes impractical for parameter exploration and inverse calibration, a setting where machine learning surrogates have been successfully used to approximate mechanistic models in systems biology, including ODE-,

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1030 DOI: 10.66693/mathai.1030
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

SDE-, and PDE-based models (Gherman et al., 2023). We contribute an operator-learning formulation for this high-dimensional symplastic growth setting.

To address this limitation, we adopt an operator-learning viewpoint. We formalize the simulator as a parameter-to-profile operator

$$\mathcal{F} : \Theta \subset \mathbb{R}^p \to \mathbb{R}^B,$$

which maps mechanical parameters to the longitudinal cell length profile obtained by spatial binning. Learning an approximation of this operator enables fast prediction without integrating the full high-dimensional state trajectory.

Beyond computational acceleration, this viewpoint clarifies structural properties of the model. The simulator defines a hybrid stochastic dynamical system whose parameter-to-profile map is locally Lipschitz away from changes in the division event schedule, while stochastic cell division induces intrinsic output variability. We show that this variability establishes an irreducible noise floor that lower-bounds the achievable surrogate error. In addition, we empirically quantify computational scaling with respect to the number of cell files and fit a power law $T(N) \approx aN^{\gamma}$ to simulation runtimes, obtaining $\gamma \approx 0.67$ with $R^2 \approx 0.93$. Although sublinear, this growth still leads to rapidly increasing cost in multi-file regimes and motivates replacing repeated ODE solves by a learned surrogate in calibration workflows.

Contributions. Our contributions are threefold. (i) We formalize symplastic leaf growth simulation as a parameter-to-profile operator of a high-dimensional hybrid stochastic system and analyze its local regularity. (ii) We establish an intrinsic stochastic error floor induced by random cell division. (iii) We empirically quantify computational scaling and demonstrate that neural surrogates provide substantial acceleration for inverse calibration while preserving profile accuracy.

2 Mathematical Model

We use the mechanical framework of Zubairova et al. (2016) for symplastic unidirectional growth of a linear leaf epidermis. The leaf is represented as a "brickwork" of cell files (rows of cells along the leaf). Each cell is characterized by *visible length* l, *relaxed length* l_r (cell wall length in the unstressed state), and *isosmotic length* l_i (related to cell biomass and osmotic content). Neighboring cells share wall fragments; the growth rate of a common fragment is determined by the sum of the free growth rates of the cells that contain it, so that cells influence each other mechanically (symplastic growth).

2.1 Biomechanics of a single cell

Following Zubairova et al. (2016), the difference between osmotic pressure inside and outside the cell is expressed in terms of lengths as

$$P_{\mathrm{osm}} = \alpha \, \frac{l_i - l}{l}, \tag{1}$$

where α is the coefficient of osmotic pressure (e.g., $\alpha = c_{\mathrm{out}} RT$ from the Van't Hoff equation). Turgor pressure (hydrostatic pressure due to wall stress) is

$$P_{\mathrm{turg}} = \beta \, \frac{l - l_r}{l_r}, \tag{2}$$

where $\beta = (S_w/S_c)E$ depends on the wall cross-section S_w, cell cross-section S_c, and Young's modulus E. Similarly to multicellular Lockhart–Ortega-type models that couple water flux and cell wall mechanics (Cheddadi et al., 2019), our formulation explicitly expresses turgor and osmotic pressure and couples growth through shared wall fragments. The yield threshold P_c and irreversible wall growth l_r belong to the cell wall extensibility tradition (Cosgrove, 1993). Water flows into the cell when the water potential $\Psi_w = (P_{\mathrm{osm}} - P_{\mathrm{osm}}^{\mathrm{out}}) - P_{\mathrm{turg}} > 0$. The growth rate of the visible cell length is proportional to this flow. In unidirectional growth, with hydraulic conductivity L_w and cell width r,

$$\frac{dl}{dt} = r \cdot l \cdot L_w \cdot (P_{\mathrm{osm}} - P_{\mathrm{turg}}), \tag{3}$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1030 DOI: 10.66693/mathai.1030
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

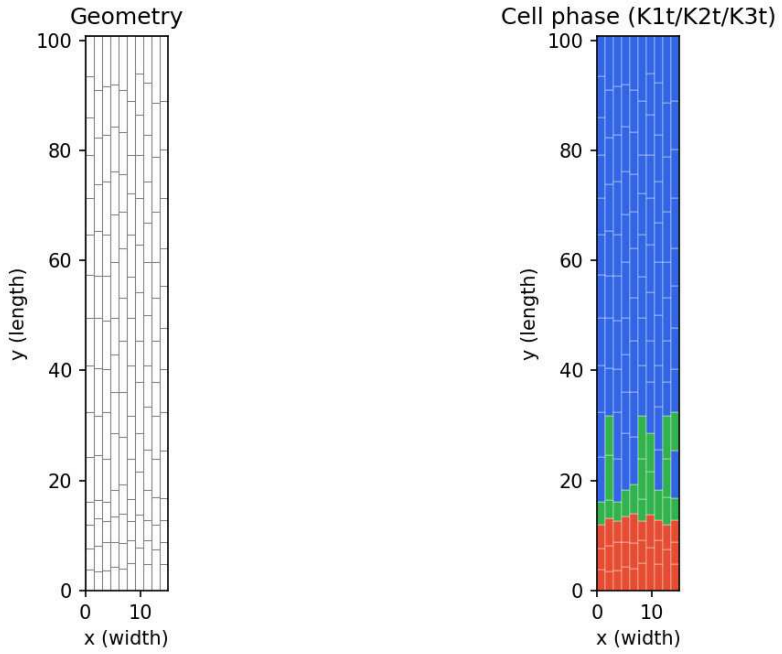


Figure 1: Multi-file leaf structure: cell files (rows along the leaf), shared transverse fragments (segments), and cell cycle phases along the longitudinal axis. Phase colors: K1t (red), K2t (green), K3t (blue).

or in specific form

$$\frac{1}{l}\frac{dl}{dt} = r \cdot L_w \cdot \left(\alpha \frac{l_i - l}{l} - \beta \frac{l - l_r}{l_r}\right). \tag{4}$$

Cell wall growth (irreversible change of relaxed length) is triggered when turgor exceeds a threshold P_c:

$$\frac{dl_r}{dt} = \begin{cases} 0, & \text{if } P_{\text{turg}} \le P_c \\ \eta \frac{dl_i}{dt} (P_{\text{turg}} - P_c)^3, & \text{otherwise,} \end{cases} \tag{5}$$

where η is the coefficient relating cell wall growth rate to biomass growth rate. The isosmotic length $l_i(t)$ is given by an explicit function of time (autonomous growth), e.g., piecewise-linear in the division zone (DZ) and elongation zone (EZ) with rates a_1, a_2 and critical value $l_{i,\max}$ (Zubairova et al., 2016).

2.2 Symplastic growth: fragments and cell files

The leaf epidermis is divided into transverse *fragments* (segments) indexed by k; fragment k has length λ_k and is shared by one cell from each file. The growth rate of the fragment is the average of the free specific growth rates of the cells that contain it:

$$\frac{d\lambda_k}{dt} = \frac{\lambda_k}{N} \sum_{n=1}^{N} \left(\frac{1}{l_{mn}} \frac{dl_{mn}}{dt}\right)_{\text{free}}, \quad (m : \lambda_k \in l_{mn}), \tag{6}$$

where N is the number of cell files, l_{mn} is the visible length of cell m in file n, and the "free" rate is given by the right-hand side of equation 4 for that cell. The visible length of each cell is the sum of the lengths of the fragments belonging to it, so dl_{mn}/dt is the sum of the corresponding $d\lambda_k/dt$. This couples all cells that share fragments and yields the symplastic dynamics. The multi-file geometry and cell phases along the leaf are illustrated in Figure 1.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1030 DOI: 10.66693/mathai.1030
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

2.3 CELL DIVISION IN THE DIVISION ZONE

Cells progress through three phases along the leaf: K1t (growth toward the division threshold), K2t (division zone, where l_i may reach the critical value), and K3t (elongation, post-division). A cell in the division zone divides when its isosmotic length l_i reaches a critical value $l_{i,\max}$. The mother cell is replaced by two daughter cells with length ratio $d/(1-d)$, where d is the division factor (random, truncated normal in $(0.1, 0.9)$ with mean 0.5). The initial isosmotic lengths of the daughters are $d \cdot l_{i,\max}$ and $(1-d) \cdot l_{i,\max}$, and their birth time t_0 is set to the division time (Zubairova et al., 2016). Segment (fragment) state is updated so that every segment reflects the new cell layout. This coupling between many cells and segments makes the multi-file regime both biologically relevant and numerically expensive. Because d is stochastic, the same parameter vector can yield different (L, t_f) across runs; in practice one may average over several runs with different random seeds for each $\boldsymbol{x}$, or include the seed as an extra input to the surrogate. The present work uses a Python reimplementation of this framework (with the same equations and division rule) to generate training data for the operator surrogate.

2.4 STATE DIMENSION AND COMPUTATIONAL COST: THE ROLE OF THE NUMBER OF CELL FILES

A central structural parameter of the model is the *number of cell files* N: the leaf epidermis is represented as N parallel rows of cells along the longitudinal axis, and each transverse fragment (segment) is shared by exactly one cell from each file, so N cells are mechanically coupled at each fragment. For N cell files and M cells per file (at a given time), the state comprises $3NM$ cell-level variables (visible length l, relaxed length l_r, and isosmotic length l_i per cell) plus $K(N, M)$ fragment lengths λ_k, where K grows with the number of cells and their connectivity along the leaf. Thus $D(N, M) \sim 3NM + K(N, M)$. The dependence on N is twofold: (i) the number of cell variables scales as NM, and (ii) each fragment couples N cells, so the interaction structure and the number of nonzero entries in the Jacobian grow with N. As a result, numerical integration cost increases rapidly with the number of cell files. We explicitly quantify this dependence in the experiments (Sec. 4) by measuring state dimension and wall-clock time for several (N, M) configurations. This motivates learning a map from parameters to observables (e.g., the cell length profile) without integrating the full state trajectory.

2.5 PARAMETER-TO-PROFILE MAP OF THE HYBRID STOCHASTIC MODEL

Let $z(t) \in \mathbb{R}^{D(N,M)}$ denote the full simulator state, consisting of cell-level variables (l, l_r, l_i) and fragment lengths λ. For a parameter vector $x \in \Theta \subset \mathbb{R}^p$ (e.g., $x = (\alpha, \eta, P_c)$), the simulator defines a hybrid stochastic dynamical system with continuous flows and discrete state resets:

$$\dot{z}(t) = G(z(t); x), \qquad z(0) = z_0(x), \tag{7}$$

where discrete events correspond to cell division in the division zone. Division events are triggered when the isosmotic length of a cell reaches a threshold; after division, the state is updated via a reset map that depends on a random division factor d.

Let ω denote a realization of the simulator randomness (e.g., a random seed determining division factors). For each (x, ω), the simulator produces a terminal state $z(t_f(x, \omega))$, where $t_f(x, \omega)$ is the stopping time at which the total leaf length first reaches the prescribed target L_y.

We define the profile projection operator

$$\mathcal{P} : \mathbb{R}^{D(N,M)} \to \mathbb{R}^B,$$

which partitions the longitudinal axis into B spatial bins and returns the corresponding mean cell lengths. The simulator thus induces a stochastic parameter-to-profile map

$$F_\omega : \Theta \to \mathbb{R}^B, \qquad F_\omega(x) = \mathcal{P}\big(z(t_f(x, \omega); x, \omega)\big). \tag{8}$$

In practice, experimental measurements correspond to averaged cell length profiles. We therefore consider the mean profile operator

$$\bar{F}(x) = \mathbb{E}_\omega \left[F_\omega(x)\right], \tag{9}$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1030 DOI: 10.66693/mathai.1030
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

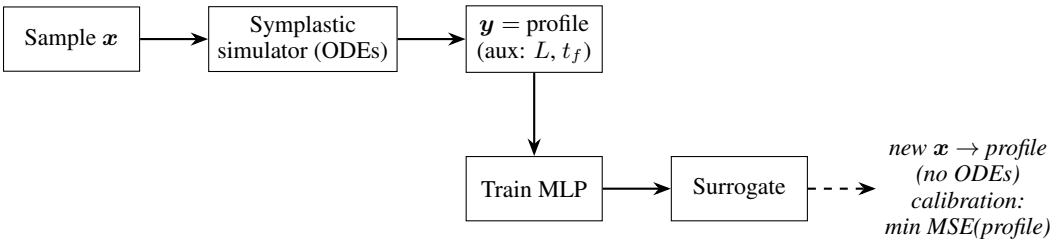


Figure 2: Operator learning pipeline. Top: sample $\boldsymbol{x} \in \Theta$, run the symplastic simulator; each run yields the profile (and auxiliary scalars L, t_f). The operator $\mathcal{F}$ maps parameters to the profile only. Bottom: train an MLP to approximate $\mathcal{F}$, then use $\widehat{\mathcal{F}}$ for fast prediction and calibration (minimize MSE between predicted and experimental profile) without solving the ODEs.

which maps mechanical parameters to the expected longitudinal profile.

This formulation makes explicit that surrogate learning therefore amounts to approximating the parameter-to-profile operator of a high-dimensional hybrid stochastic dynamical system.

3 Operator Learning and Surrogate

We approximate the parameter-to-profile operator $\mathcal{F} : \Theta \subset \mathbb{R}^p \to \mathbb{R}^B$ mapping parameters to the longitudinal profile only; Θ is the set of admissible mechanical parameters (e.g., α, η, P_c). The simulator also yields auxiliary scalar outputs L (final leaf length) and t_f (stopping time), which we may predict alongside the profile but are not part of the operator $\mathcal{F}$. The goal is to learn an approximation $\widehat{\mathcal{F}}$ to $\mathcal{F}$ from data $\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}$ with $\boldsymbol{y}^{(i)} = \mathcal{F}(\boldsymbol{x}^{(i)})$, so that we can evaluate $\widehat{\mathcal{F}}(\boldsymbol{x})$ without integrating the ODEs.

3.1 Pipeline

The pipeline is summarized in Figure 2. Parameter vectors $\boldsymbol{x} \in \Theta$ are sampled and fed to the symplastic simulator; each run yields the profile $(\bar{\ell}_1, \ldots, \bar{\ell}_B) \in \mathbb{R}^B$ and auxiliary scalar outputs (L, t_f). The surrogate approximates $\mathcal{F}$ mapping $\boldsymbol{x} \to$ profile; the MLP may also predict L and t_f as auxiliary targets. The trained surrogate $\widehat{\mathcal{F}}$ then predicts the profile for new $\boldsymbol{x}$ without solving the ODEs. The main application is calibration: given an experimental profile binned to B bins, we minimize the MSE between $\widehat{\mathcal{F}}(\boldsymbol{x})$ and the experimental profile over $\boldsymbol{x} \in \Theta$.

3.2 Inputs and outputs

We vary a small set of parameters; the rest (geometry, etc.) are fixed. The input vector $\boldsymbol{x}$ typically includes the law-of-growth parameters:

- α (coefficient of osmotic pressure, Eq. 1),
- η (coefficient of cell wall growth rate, Eq. 5),
- P_c (turgor threshold when wall growth begins, Eq. 5).

The set of calibratable parameters can be extended to include parameters of the growth function $l_i(t)$ (e.g., cell cycle time, elongation phase duration, division and elongation thresholds, min/max cell sizes). These are among the parameters with significant sensitivity (Zubairova et al., 2016). Each run of the simulator returns a final total leaf length L, the time t_f at which the target length L_y was reached, and a *cell length profile*: the leaf is divided into B bins along the longitudinal axis (distance from the base), and the mean cell length in each bin is recorded. The operator $\mathcal{F}$ maps to the profile $(\bar{\ell}_1, \ldots, \bar{\ell}_B) \in \mathbb{R}^B$; L and t_f are auxiliary simulator outputs. Experimental data (cell lengths vs. distance from the leaf base) can be binned into the same B bins to obtain a target profile $\boldsymbol{y}_{\text{exp}}^{\text{prof}}$. Calibration amounts to finding parameters $\boldsymbol{x}$ that minimize the MSE between the surrogate-predicted profile and $\boldsymbol{y}_{\text{exp}}^{\text{prof}}$, without running the full ODE solver in the inner loop.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1030 DOI: 10.66693/mathai.1030
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

3.3 Training data

We draw N parameter vectors $\boldsymbol{x}^{(i)}$ from uniform distributions over prescribed bounds, run the simulator for each, and collect successful pairs $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$. Failed runs (numerical errors or non-convergence) are discarded. The dataset is $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{n}$. We split $\mathcal{D}$ into training (e.g. 80%) and test (20%) sets; surrogates are trained only on the training set and evaluated on the test set to report MAE and RMSE without leakage.

3.4 MLP operator approximation

A multi-layer perceptron (MLP) regressor is trained on $\mathcal{D}$ to approximate the operator $\mathcal{F}$: it maps $\boldsymbol{x} \in \Theta$ to the predicted profile $\widehat{\mathcal{F}}(\boldsymbol{x}) \in \mathbb{R}^B$ (and may optionally predict L, t_f as auxiliary outputs). We use scikit-learn's `MLPRegressor` with two hidden layers (e.g., 32 units each) and early stopping. The MLP is fast at test time, which is essential for calibration: the optimizer (e.g., L-BFGS-B) repeatedly evaluates the loss (MSE between predicted and experimental profile) and thus calls $\widehat{\mathcal{F}}$ many times without running the ODE solver.

Local regularity. Proposition 1 (Piecewise local Lipschitz continuity). Assume that for a fixed (x_0, ω) the event schedule (the number and order of cell divisions) does not change in a neighborhood $U \subset \Theta$ of x_0, and that the vector field $G(\cdot; x)$ is locally Lipschitz in z and continuous in x. Assume further that the reset maps at division events are locally Lipschitz in the pre-event state.

Then there exists a constant $L_{x_0,\omega}$ such that for all $x_1, x_2 \in U$,

$$\|F_\omega(x_1) - F_\omega(x_2)\| \leq L_{x_0,\omega}\|x_1 - x_2\|.$$

Sketch of argument. Stability of ODE flows under Lipschitz vector fields implies continuous dependence of trajectories on parameters. Since reset maps are Lipschitz and the event schedule is fixed, the full hybrid flow remains locally Lipschitz. The projection $\mathcal{P}$ preserves Lipschitz continuity.

Remark. Changes in the event schedule may occur near parameter values where division thresholds are reached at different times. Such regions introduce non-smoothness in the parameter-to-profile map and increase approximation difficulty.

Intrinsic stochastic error floor. Proposition 2 (Irreducible error due to stochastic division). Let $Y = F_\omega(x) \in \mathbb{R}^B$ be the stochastic profile at fixed x, with mean $\mu(x) = \mathbb{E}[Y]$ and covariance matrix $\Sigma(x)$.

For any deterministic predictor $g : \Theta \to \mathbb{R}^B$,

$$\mathbb{E}_\omega\|g(x) - Y\|_2^2 = \|g(x) - \mu(x)\|_2^2 + \operatorname{tr}\Sigma(x) \geq \operatorname{tr}\Sigma(x).$$

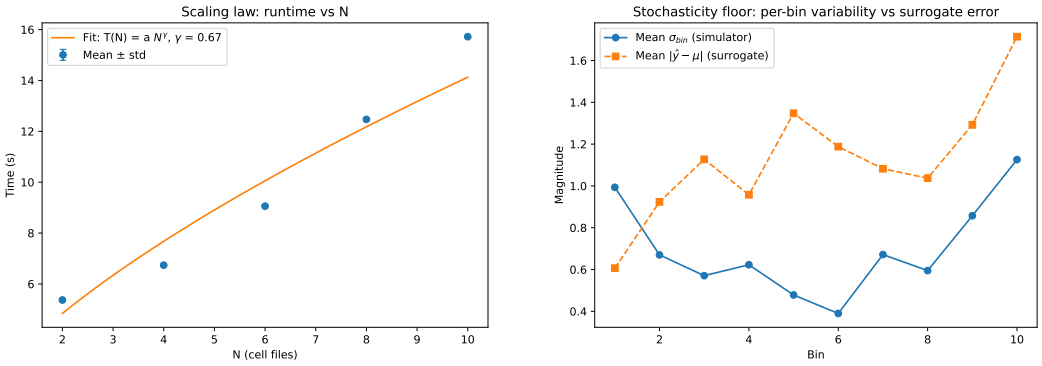
Therefore, the trace of the output covariance acts as an intrinsic noise floor: no deterministic surrogate can achieve mean squared error below the variance induced by stochastic cell division.

Implications for surrogate learning. Proposition 1 suggests that away from event-schedule transitions, the parameter-to-profile map is locally regular, which justifies approximation by smooth function classes such as MLPs. Proposition 2 shows that even a perfect surrogate cannot achieve error below the intrinsic stochastic variance of the simulator. Together, these results clarify both the feasibility and the fundamental limits of neural operator approximation in this hybrid stochastic setting.

4 Experiments

We approximate the mean parameter-to-profile operator $\bar{\mathcal{F}} : \Theta \to \mathbb{R}^B$ using a neural surrogate trained on simulator-generated samples.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1030 DOI: 10.66693/mathai.1030
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026



(a) Scaling: runtime vs N and power-law fit ($\gamma \approx 0.67$, $R^2 \approx 0.93$).

(b) Stochasticity floor: surrogate RMSE vs. within-simulator σ.

Figure 3: (a) Simulation runtime vs number of cell files N. (b) Surrogate profile RMSE vs. within-simulator variability for fixed parameter vectors (ratio ≤ 1: surrogate within stochastic spread).

4.1 Setup

We use the multi-file configuration: $N = 10$ cell files, $M = 4$ cells per file initially, cell division enabled, target length $L_y = 100$ (in model units). Each simulation returns the profile ($B = 10$ spatial bins) and auxiliary scalars L and t_f. Parameter ranges: $\alpha \in [5, 15]$, $\eta \in [0.05, 0.3]$, $P_c \in [1, 4]$ bar (Zubairova et al., 2016). The dataset consists of $n = 200$ successful runs, split 80/20 into train and test. For each parameter vector we use a single simulator run (one random seed). We train an MLP and, for comparison, a Ridge regression baseline on the same inputs and outputs.

4.2 Simulator scaling: dependence on the number of cell files

Because the number of cell files N is a key structural parameter (state dimension and fragment coupling both scale with N), we explicitly measure how state dimension D and wall-clock time per simulation depend on N (and on M). We run the simulator for several configurations (N, M) (e.g., $(1, 4)$, $(2, 4)$, and optionally $(2, 8)$ or $(4, 4)$) and record the state dimension (number of cell-level variables plus fragment lengths) and mean runtime. State dimension D and mean runtime for each configuration are reported in Table 1; we discuss the observed scaling with N (and with D). This dependence justifies the use of operator learning: when N or M grows, repeated simulation becomes prohibitive for calibration and parameter studies.

We further summarize the empirical scaling by fitting a power law $T(N) \approx aN^{\gamma}$ to the runtimes in Table 1 via log–log regression (using the reported times for $N = 2, 4, 6, 8, 10$). This yields $\gamma \approx 0.67$ with $R^2 \approx 0.93$, indicating a sublinear but steadily increasing computational cost with the number of coupled cell files (Figure 3a). This motivates replacing repeated ODE solves by the learned surrogate in parameter studies and calibration.

4.3 Profile prediction quality

We evaluate the surrogate on the held-out test set. On the test set we obtain MAE (RMSE) of 20.29 (24.23) for L, 5.76 (6.73) for t_f, and 1.23 (1.50) for the binned profile. We also report R^2 for the profile and maximum absolute error across bins. We plot 5–10 example profiles (true vs. predicted) to assess shape fidelity. Permutation feature importance for α, η, P_c on the profile indicates that α has the strongest influence, followed by P_c and η. If a baseline is trained, we include its metrics in the same table.

4.4 Stochasticity of the simulator

Because cell division is stochastic, the same parameter vector $\boldsymbol{x}$ yields different profiles across runs. For five fixed parameter vectors $\boldsymbol{x}$ we run the simulator with $S = 10$ different seeds and

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1030 DOI: 10.66693/mathai.1030
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

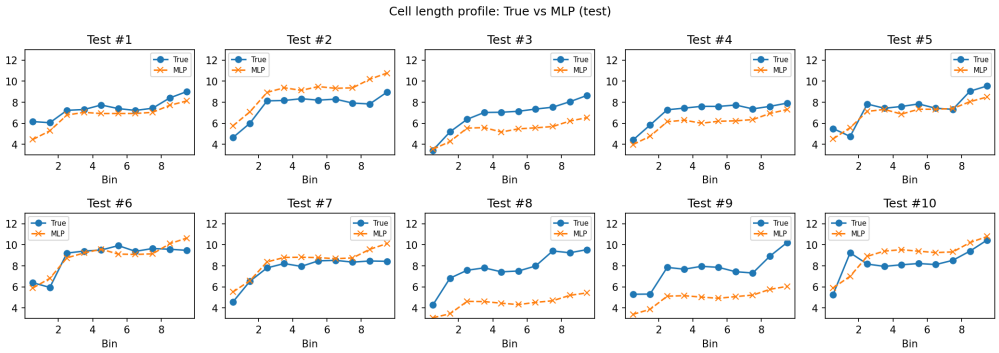


Figure 4: Example cell length profiles: true (from simulator) vs. MLP-predicted on the test set. Each subplot is one test configuration.

compute the per-bin standard deviation of the profile. We compare the surrogate's test RMSE to this within-simulator variability: when the surrogate error is comparable to or smaller than the typical run-to-run spread, the approximation is adequate for calibration. This empirical variability provides an estimate of the intrinsic stochastic noise floor described in Proposition 2. Figure 3b illustrates this comparison for our setup.

4.5 Calibration: synthetic inverse test

We perform a synthetic inverse test: choose a "true" parameter vector $\boldsymbol{x}^*$, generate a target profile $\boldsymbol{y}^*$ (one run or mean over a few runs), add small Gaussian noise to mimic measurement error, and run calibration by minimizing MSE between the surrogate-predicted profile and the noisy target over $\boldsymbol{x}$. We report the parameter error $|\hat{\boldsymbol{x}} - \boldsymbol{x}^*|$ (per component or norm), the profile MSE at the recovered $\hat{\boldsymbol{x}}$, the number of loss evaluations, and wall-clock time. We compare with the cost of calibration using the full simulator when feasible (e.g., few iterations) and report speedup. Figure 5a shows an example: target, noisy target, and profile after calibration.

This demonstrates that surrogate-based calibration can recover parameters within the intrinsic stochastic variability of the model.

4.6 Computational efficiency

We report: (i) mean wall-clock time for one full simulation; (ii) time for 10^4 surrogate evaluations; (iii) time for one calibration run (surrogate-only). In our setup, one simulator run for $(N, M) = (2, 4)$ takes on the order of 4 s, while 10^4 surrogate evaluations take on the order of 5 ms, yielding a speedup of $\sim 10^6$ per evaluation in our setup; one calibration run takes on the order of 15 ms. The speedup factor and the cost of calibration with the surrogate summarize the practical benefit of the learned operator.

4.7 Results summary

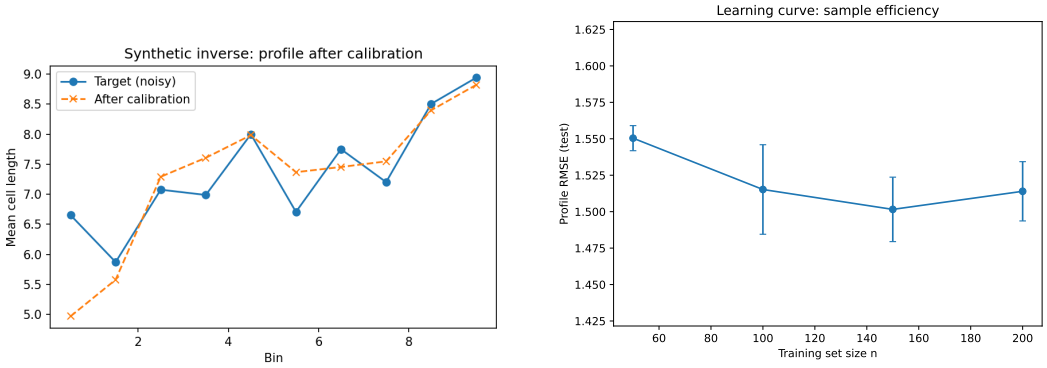
State dimension D and mean runtime per simulation for configurations (N, M) from $(2, 4)$ to $(10, 4)$ are reported in Table 1; the fitted power law yields $\gamma \approx 0.67$ with $R^2 \approx 0.93$ (see Figure 3a), confirming that computational cost grows with the number of cell files and motivating the use of a surrogate for parameter studies. Test-set metrics for the MLP are MAE (RMSE) 20.29 (24.23) for L, 5.76 (6.73) for t_f, and 1.23 (1.50) for the profile. Example true vs. predicted profiles are shown in Figure 4. The stochasticity floor is illustrated in Figure 3b; synthetic inverse calibration and the learning curve are shown in Figure 5a and Figure 5b, supporting the use of larger datasets or active learning in future work.

Profile prediction accuracy is sufficient for calibration: the surrogate's profile RMSE is comparable to or below the within-simulator variability (stochasticity floor) when the same parameters are run with different division seeds, consistent with Proposition 2. The synthetic inverse test shows that

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1030 DOI: 10.66693/mathai.1030
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Table 1: Dependence on the number of cell files N and cells per file M: state dimension D and mean wall-clock time per simulation. N is the number of cell files (parallel rows); each fragment couples N cells.

(N, M)	**State dim.** D	**Time (s)**
(2, 4)	55	5.37
(4, 4)	106	6.74
(6, 4)	160	9.06
(8, 4)	211	12.47
(10, 4)	264	15.72



(a) Synthetic inverse: target, noisy target, and profile at $\hat{\boldsymbol{x}}$.

(b) Profile RMSE vs. training set size.

Figure 5: (a) Synthetic inverse test: profile after calibration. (b) Learning curve: profile RMSE decreases with training set size, supporting active learning.

parameters can be recovered by minimizing MSE between the surrogate-predicted profile and a noisy target, with the recovered profile matching the target within the intrinsic stochastic spread. The learned surrogate is orders of magnitude faster than one ODE solve per evaluation (speedup $\sim 10^6$ per evaluation in our setup), so calibration over $\boldsymbol{x} \in \Theta$ is feasible in practice without running the full simulator in the inner loop.

5 Conclusion

We studied operator learning for a high-dimensional symplastic leaf growth model with stochastic cell division (Zubairova et al., 2016). The state dimension scales as $D(N, M) \sim 3NM + K(N, M)$, and fragment-based mechanical coupling leads to dense interactions, making repeated numerical simulation increasingly costly as the number of cell files grows.

In line with surrogate-modelling practice for biological systems (Gherman et al., 2023), we justified the surrogate approach by explicitly quantifying computational scaling with respect to the number of cell files and by accounting for the intrinsic stochasticity floor induced by random cell division. We formalized the simulator as a nonlinear operator $\mathcal{F} : \Theta \to \mathbb{R}^B$ mapping mechanical parameters to the longitudinal cell length profile, and trained an MLP surrogate to approximate $\mathcal{F}$.

The learned surrogate replaces repeated ODE integration with fast forward evaluation and enables efficient calibration to experimental profiles by minimizing MSE over the parameter space. We reported generalization performance on held-out configurations and analyzed feature importance to assess parameter influence on the predicted profiles. Overall, our results indicate that neural operator approximation provides a computationally scalable framework for accelerating analysis and inverse modeling of coupled high-dimensional biological growth dynamics.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1030 DOI: 10.66693/mathai.1030
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 28 May 2026 · Published: 28 May 2026

Consistent with recommendations from surrogate-modelling practice (Gherman et al., 2023), natural next steps include active learning to add training points in regions of higher surrogate error, dimensionality reduction of inputs or outputs to improve data efficiency, and more explicit treatment of stochasticity (e.g., averaging over division factors or incorporating the random seed as an input). Further directions include optimization over the surrogate (e.g., maximizing L for fixed time (Salehi et al., 2020)) and validation on real experimental profiles.

Code and reproduction instructions are available upon request.

REFERENCES

Ibrahim Cheddadi, Michel Génard, Nadia Bertin, and Christophe Godin. Coupling water fluxes with cell wall mechanics in a multicellular model of plant development. *PLoS Computational Biology*, 15(6):e1007121, 2019. doi: 10.1371/journal.pcbi.1007121.

Daniel J. Cosgrove. Wall extensibility: its nature, measurement and relationship to plant cell growth. *New Phytologist*, 124(1):1–23, 1993. doi: 10.1111/j.1469-8137.1993.tb03795.x.

Ioana M. Gherman, Zahraa S. Abdallah, Wei Pang, Thomas E. Gorochowski, Claire S. Grierson, and Lucia Marucci. Bridging the gap between mechanistic biological models and machine learning surrogates. *PLoS Computational Biology*, 19(4):e1010988, 2023. doi: 10.1371/journal.pcbi.1010988.

Mina Salehi, Saeed Farhadi, Ahmad Moieni, Naser Safaie, and Reza Ahmadi. Mathematical modeling of growth and paclitaxel biosynthesis in corylus avellana cell culture responding to fungal elicitors using multilayer perceptron-genetic algorithm. *Frontiers in Plant Science*, 11:1148, 2020. doi: 10.3389/fpls.2020.01148.

Ulyana Zubairova, Sergey Nikolaev, Aleksey Penenko, Nikolay Podkolodnyy, Sergey Golushko, Dmitry Afonnikov, and Nikolay Kolchanov. Mechanical behavior of cells within a cell-based model of wheat leaf growth. *Frontiers in Plant Science*, 7:1878, 2016. doi: 10.3389/fpls.2016.01878. Front. Plant Sci. 7:1878.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1032 DOI: 10.66693/mathai.1032
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

DYNAMIC DATA CLASSIFICATION BASED ON A SEMI-SUPERVISED LOCAL POISSON LABEL PROPAGATION METHOD

Constantin Ruchkin

"CR Consult", Donetsk, Russia

construchk@gmail.com

ABSTRACT

We study semi-supervised classification in a dynamic data-stream setting, where objects and their relations evolve over time while only a small fraction of observations is labeled. Classical graph-based semi-supervised learning methods, such as label propagation and Laplacian-based regularization, typically reduce learning to the solution of a global graph problem. This requires storing the full graph and recomputing the solution whenever the graph structure changes, which becomes computationally expensive in streaming environments, especially when noisy, corrupted, or obsolete observations must be removed promptly from the model. Moreover, classical harmonic formulations degenerate in extremely low-label regimes.

We propose *Semi-Supervised Local Poisson Label Propagation* (SSLPLP), a local Poisson-based framework for evolving graphs. The method formulates prediction updates through a graph Poisson equation with class-dependent sources and sinks induced by labeled vertices, aggregated through class supernodes. Instead of maintaining the full graph, SSLPLP keeps only a compact active neighborhood, where each newly arriving observation is connected to a limited set of active neighbors within a temporal window or k-NN structure.

The key efficiency mechanism is local graph reduction via the star–mesh transformation (Kron reduction / Schur complement). We prove that this reduction is exact: under the zero-sum solvability condition, elimination of zero-forcing unlabeled vertices preserves Poisson potentials on the active region. We further prove linear convergence of the iterative Poisson solver on the reduced graph, derive its spectral rate, and bound the numerical error accumulated over sequential reductions. Computational complexity is $O(\tau^2 C)$ per streaming step, where τ is the active window size and C the number of classes, compared with $\Omega(n\tau^2 C)$ for batch recomputation.

We validate SSLPLP on two datasets: synthetic Two Moons, ECG arrhythmia classification (INCART-ECG). On temporally ordered streams, SLPLP achieves 88–96% accuracy with as few as 2–5 labeled examples per class, consistently outperforming quantized label propagation and labels-only baselines. The framework is particularly suitable for sparse-label regimes with local temporal structure and naturally accommodates concept drift through exponentially decaying edge weights.

1 INTRODUCTION

Graph-based semi-supervised learning (GSSL) is based on the assumption that similar objects connected in a similarity graph should receive similar labels. Classical methods such as harmonic label propagation, graph Laplacian regularization, and manifold-based techniques construct a graph over labeled and unlabeled data and infer class information by enforcing smoothness of a labeling function over that graph Zhu et al. (2003); Zhou et al. (2004); Belkin et al. (2006). These methods have

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1032 DOI: 10.66693/mathai.1032
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

shown strong performance in static settings where the full dataset is available in advance. However, in many practical applications data arrive sequentially and the underlying graph changes over time. Examples include biomedical signals, sensor streams, event logs, video streams, and evolving interaction networks. In such scenarios, the learner must update predictions online while satisfying strict memory and time constraints. Classical Laplacian-based methods are poorly adapted to this regime because they require storing a large graph and repeatedly recomputing a global solution whenever new vertices or edges are added or old observations are removed.

This limitation becomes even more severe when the fraction of labeled samples is very small. In low-label regimes, standard Laplacian formulations may degenerate and fail to propagate class information effectively across the unlabeled population Nadler et al. (2009). This motivated a line of work on alternative graph-based formulations, including p-Laplacian methods, properly weighted Laplacians, Lipschitz learning, and especially Poisson learning, where labeled data are injected as sources and sinks rather than fixed boundary conditions El Alaoui et al. (2016); Slepčev and Thorpe (2019); Calder et al. (2020). Poisson-based formulations are particularly appealing when label information is scarce, since they avoid some of the degeneracies of classical harmonic extension.

Recent work on streaming GSSL introduced temporal graph constructions and online graph compression. In particular, Temporal Label Propagation (TLP) of Wagner et al. Wagner et al. (2018) showed that star–mesh elimination can preserve the *harmonic* solution on the active temporal region while keeping only a compact graph summary. On the INCART-ECG benchmark, TLP achieves 94.9% accuracy with only 2 labels per class, and its per-step time is independent of the stream length n. This suggests that a similar compression idea may be transferred from harmonic propagation to Poisson-based learning—and that the transfer yields an algorithm better suited to extremely low-label regimes.

In this work, we propose *Semi-Supervised Local Poisson Label Propagation* (SSLPLP), combining three principles: (i) a temporal or local active neighborhood for streaming data, (ii) class supernodes aggregating labeled information as equivalent source terms, and (iii) local graph reduction through star–mesh elimination of obsolete unlabeled vertices. We make the following concrete contributions:

Algorithm. SSLPLP maintains a compact weighted graph H_n of size $\tau + C$ over a stream of length n, processing each point in $O(\tau^2 C + m_n)$ time and using $O(\tau^2 \log(n\omega) + (m_n + \tau)\log\chi)$ bits, where m_n is the total number of labeled points seen so far, ω the ratio of maximum to minimum similarity values, and χ the domain size. **Equivalence statement.** The Poisson solution on H_n is identical to the Poisson solution on the full temporal vicinity graph $G_\tau^{(n)}$. **Convergence statement.** The iterative solver converges linearly at rate $\rho = \|D_n^{-1} W_n\|_2 < 1$, with an explicit spectral bound. **Numerical error bound.** After k sequential reductions, the accumulated floating-point error is bounded by $O(k\,\varepsilon_{\mathrm{mach}}\,\kappa(L))$, where $\kappa(L)$ is the condition number of the active Laplacian. **Experiments.** Validation on two datasets with ablation studies confirming the benefit of star–mesh reduction in streaming settings.

2 Related Work

Classical graph SSL. Zhu and Ghahramani introduced iterative label propagation Zhu and Ghahramani (2002); Zhu, Ghahramani, and Lafferty established the Gaussian fields and harmonic functions formulation Zhu et al. (2003). Zhou et al. proposed local-and-global consistency Zhou et al. (2004). Manifold regularization was developed by Belkin, Niyogi, and Sindhwani Belkin et al. (2006). Bengio et al. unified several propagation methods through a quadratic-criterion perspective Bengio et al. (2006).

Low-label regime. Nadler, Srebro, and Zhou showed that graph Laplacian methods degenerate when the number of unlabeled samples grows while the number of labels remains fixed Nadler et al. (2009). This motivated p-Laplacian methods El Alaoui et al. (2016); Slepčev and Thorpe (2019), the game-theoretic p-Laplacian Calder (2019), properly weighted Laplacians Calder and Slepčev (2020), and Poisson learning Calder et al. (2020) where labels are injected as sources and sinks. Avrachenkov et al. studied regularized Laplacian methods Avrachenkov et al. (2017).

Streaming and scalable SSL. Valko et al. proposed online SSL on quantized graphs Valko et al. (2010). Ravi and Diao developed a streaming approximation framework Ravi and Diao (2016).

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1032 DOI: 10.66693/mathai.1032
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Wagner et al. introduced Temporal Label Propagation (TLP) for data streams, using star–mesh elimination to maintain the exact harmonic solution on a compact active region Wagner et al. (2018). Viswanathan et al. studied SSL with multiple graphs Viswanathan et al. (2019); Sharma and Jones considered graph learning for SSL Sharma and Jones (2023).

SSLPLP is positioned at the intersection of these directions. From classical GSSL it inherits smoothness-based propagation. From low-label Poisson learning it inherits the source–sink formulation. From TLP it inherits the idea of maintaining a compact active graph under continual updates, while replacing harmonic propagation with Poisson propagation to address the low-label degeneracy observed in harmonic methods.

3 Problem Formulation

We consider a stream of observations

$$x_1, x_2, \ldots, x_n, \ldots, \qquad x_t \in \mathcal{X},$$

arriving sequentially in time. A small subset of these observations is labeled, while the remaining majority is unlabeled. Let $L_n \subseteq \{1, \ldots, n\}$ denote the indices of labeled points observed up to time n, and let $y_i \in \{1, \ldots, C\}$, $i \in L_n$, be the corresponding class labels.

We assume that pairwise similarity is given by a symmetric nonnegative function $\mathrm{Sim}(x_i, x_j) \geq 0$. Based on this, one may define a dynamic graph $G^{(n)} = (V^{(n)}, E^{(n)}, W^{(n)})$ where the vertex set contains all observations up to time n. However, storing and updating the full graph is infeasible for long streams.

We introduce an *active region* $A_n \subseteq \{1, \ldots, n\}$ consisting only of currently relevant vertices, defined through a temporal window, a local kNN graph, or a hybrid locality rule. Labeled data are aggregated into *class supernodes* $s_1, \ldots, s_C$, one per class.

Learning is formulated through the graph Poisson equation

$$L_H U = B,$$

where H is the reduced active graph, L_H its graph Laplacian, U the matrix of class potentials, and B the source term. The source is nonzero only at supernodes:

$$B(s_c, :) = m_c\big(e_c - \bar{y}\big), \qquad \bar{y} = \tfrac{1}{m} \textstyle\sum_{i \in L_n} e_{y_i},$$

and $B(v, :) = 0$ for every active unlabeled node v. The prediction rule is $\hat{y}(x) = \arg\max_c U_c(x)$.

4 A Local Semi-Supervised Poisson Label Propagation Method

4.1 Active Graph Representation

At time step n, we maintain a compact reduced graph $H_n = (V_n, E_n, W_n)$ with $V_n = Q_n \cup S$, where Q_n is the set of active unlabeled vertices (the latest τ observations by default) and $S = \{s_1, \ldots, s_C\}$ is the set of class supernodes.

4.2 Class Supernodes and Source Construction

For each class c, supernode s_c aggregates the influence of all labeled observations with label c. The edge weight from s_c to an active unlabeled node $v \in Q_n$ is maintained as

$$w(s_c, v) = \sum_{i \in L_n^{(c)}} \mathrm{Sim}(x_i, v),$$

where $L_n^{(c)} = \{i \in L_n : y_i = c\}$. When a new labeled point x_n of class c arrives, the supernode weights are updated incrementally:

$$w(s_c, v) \leftarrow w(s_c, v) + \mathrm{Sim}(x_n, v), \qquad \forall\, v \in Q_n.$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1032 DOI: 10.66693/mathai.1032
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

The source matrix satisfies

$$B_n(s_c,:) = m_c\big(e_c - \bar{y}^{(n)}\big), \qquad \bar{y}^{(n)} = \frac{1}{m_n}\sum_{i\in L_n} e_{y_i},$$

and $B_n(v,:) = 0$ for all $v \in Q_n$. One verifies that $\sum_{c=1}^{C} B_n(s_c,:) = 0$, so the zero-sum solvability condition holds.

4.3 Dynamic Update Rule

Case 1: x_n is labeled with class c. Update $m_c \leftarrow m_c + 1$, $m_n \leftarrow m_n + 1$, recompute $\bar{y}^{(n)}$, and update $w(s_c, v)$ for all $v \in Q_n$. Recompute the source matrix B_n.

Case 2: x_n is unlabeled. Insert x_n into Q_n. Set edge weights:

$$w(x_n,u) = \mathrm{Sim}(x_n,u), \quad u \in Q_{n-1}; \qquad w(x_n,s_c) = \sum_{i\in L_n^{(c)}} \mathrm{Sim}(x_n,x_i).$$

If $|Q_n| > \tau$, eliminate the oldest unlabeled vertex x_o via star–mesh reduction.

4.4 Local Graph Reduction via Star–Mesh Elimination

When an obsolete unlabeled vertex x_o is evicted, it is eliminated by the star–mesh transformation (equivalently Kron reduction / Schur complement). Let $\mathcal{N}(x_o)$ denote its neighbors in H_n, which may include both active unlabeled vertices and class supernodes s_c.

General weighted formula. For every pair $i, j \in \mathcal{N}(x_o)$, $i \neq j$:

$$w_{ij} \leftarrow w_{ij} + \frac{w_{i,o}\, w_{o,j}}{d_o}, \qquad d_o = \sum_{k\in\mathcal{N}(x_o)} w_{o,k}. \tag{1}$$

In particular, when x_o has an edge to supernode s_c, the elimination propagates labeled information into the active graph:

$$w_{i,s_c} \leftarrow w_{i,s_c} + \frac{w_{i,o}\, w_{o,s_c}}{d_o}, \qquad \forall\, i \in \mathcal{N}(x_o)\setminus\{s_c\}.$$

After all weight updates, vertex x_o is removed from H_n.

Matrix form. If the Laplacian is partitioned as

$$L = \begin{bmatrix} L_{TT} & L_{TO} \\ L_{OT} & L_{OO} \end{bmatrix},$$

where T denotes retained vertices (active nodes plus supernodes) and O the eliminated vertex, then the reduced Laplacian is the Schur complement:

$$L_{\mathrm{red}} = L_{TT} - L_{TO}L_{OO}^{-1}L_{OT}.$$

For a single-vertex elimination $O = \{x_o\}$, $L_{OO} = d_o$ is the scalar weighted degree, and the formula reduces to equation 1. For simultaneous elimination of multiple vertices, the matrix inverse L_{OO}^{-1} is required; in practice we eliminate one vertex per step to keep updates local.

4.5 Iterative Poisson Solver on the Reduced Graph

We adapt the iterative Poisson learning scheme of Calder et al. (2020) to the local graph H_n. With $L_{H_n} = D_n - W_n$ and source B_n, we iterate from $U_n^{(0)} = 0$:

$$U_n^{(t+1)} = D_n^{-1}\big(W_n U_n^{(t)} + B_n\big), \qquad t = 0,1,2,\ldots$$

Gauge condition. Since $\mathbf{1}^\top B_n = 0$ and $\mathbf{1}^\top L_{H_n} = 0$, the degree-weighted zero-mean condition $\mathbf{1}^\top D_n U_n^{(t)} = 0$ is preserved: if $U_n^{(0)}$ satisfies it, all iterates do.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1032 DOI: 10.66693/mathai.1032
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Stopping criterion. We stop when

$$\frac{\|U_n^{(t+1)} - U_n^{(t)}\|_F}{\max\{1, \|U_n^{(t)}\|_F\}} \le \varepsilon$$

or when $t \ge T_{\max}$. Since the graph is reduced and sparse, each iteration costs only $O(\tau^2 C)$ floating-point operations.

Optional class reweighting. Let $b = (b_1, \dots, b_C)^\top$ be the target class proportions and $\bar{y}^{(n)}$ the empirical labeled proportions. Then

$$\widetilde{U}_n = U_n \operatorname{diag}\left(\frac{b_1}{\bar{y}_1^{(n)}}, \dots, \frac{b_C}{\bar{y}_C^{(n)}}\right),$$

and the prediction uses $\widetilde{U}_n$ in place of U_n. If no prior is available, this step is omitted.

4.6 Concept Drift Adaptation

In non-stationary streams, the statistical relationship between observations and labels may change over time. SSLPLP accommodates concept drift through two mechanisms.

Decayed star–mesh reduction. When eliminating x_o, multiply existing edge weights by a forgetting factor $\alpha \in (0, 1)$ before adding the star–mesh contribution:

$$w_{ij} \leftarrow \alpha\, w_{ij} + \frac{w_{i,o}\, w_{o,j}}{d_o}. \tag{2}$$

With $\alpha < 1$, older structure is downweighted relative to recent observations. A first-order stability argument bounds the effect on Poisson potentials.

Sliding supernode weights. Instead of accumulating all labeled points, maintain supernode weights with exponential decay: when a new labeled point x_n of class c arrives,

$$w(s_c, v) \leftarrow (1 - \beta)\, w(s_c, v) + \beta\, \mathrm{Sim}(x_n, v),$$

where $\beta \in (0, 1)$ controls the learning rate of the supernode representation.

4.7 Prediction and Algorithmic Summary

After graph update, optional reduction, and iterative solution of the Poisson system, the prediction for unlabeled point x_n is

$$\hat{y}_n = \arg\max_{c=1,\dots,C} U_n(x_n, c), \quad \text{or} \quad \hat{y}_n = \arg\max_{c=1,\dots,C} \widetilde{U}_n(x_n, c).$$

5 Theoretical Remarcs

5.1 Exact Equivalence Under Schur Reduction

1)Poisson Reduction Equivalence. Let G be a connected weighted graph with Laplacian L, and let the graph Poisson problem $LU = B$ satisfy the zero-sum solvability condition $\mathbf{1}^\top B = 0$. Partition the vertex set into retained vertices T and eliminated vertices O, and suppose $B_O = 0$. Then the restriction of the global Poisson solution U^* to T equals the unique solution of the reduced system

$$L_{\mathrm{red}}\, U_T = B_T, \qquad L_{\mathrm{red}} = L_{TT} - L_{TO} L_{OO}^{-1} L_{OT},$$

subject to the induced gauge condition $\mathbf{1}_T^\top D_T U_T = 0$.

2) Streaming Equivalence. At every timestep n, the matrix U_n returned by the iterative solver of Algorithm 1 (with $\varepsilon = 0$ and $T_{\max} = \infty$) is identical to the restriction of the Poisson solution on the full temporal vicinity graph $G_\tau^{(n)}$ to the active region $V_n = Q_n \cup S$.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1032 DOI: 10.66693/mathai.1032
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Algorithm 1 SSLPLP with Iterative Local Poisson Update

Input: stream (x_n), Sim, budget τ, decay α, tolerance ε, $T_{\max}$
Initialize supernodes $s_1, \dots, s_C$; $Q_0 = \emptyset$; $m_c \leftarrow 0$ for all c; $W \leftarrow \mathbf{0}$
for each new observation x_n **do**
  if x_n is labeled with class c **then**
    $m_c \leftarrow m_c + 1$; $m_n \leftarrow m_n + 1$; update $\bar{y}^{(n)}$
    $w(s_c, v) \leftarrow w(s_c, v) + \mathrm{Sim}(x_n, v)$ for all $v \in Q_n$ ▷ supernode update
    recompute B_n
  else
    $Q_n \leftarrow Q_{n-1} \cup \{x_n\}$
    add edges: $w(x_n, u) = \mathrm{Sim}(x_n, u)$ for $u \in Q_{n-1}$
    add edges: $w(x_n, s_c) = \sum_{i \in L_n^{(c)}} \mathrm{Sim}(x_n, x_i)$
  end if
  if $|Q_n| > \tau$ **then**
    let x_o be oldest vertex in Q_n
    for each pair $i, j \in \mathcal{N}(x_o), i \neq j$ **do**
      $w_{ij} \leftarrow \alpha\, w_{ij} + w_{i,o}\, w_{o,j}/d_o$ ▷ decayed star–mesh
    end for
    $Q_n \leftarrow Q_n \setminus \{x_o\}$; remove x_o from W
  end if
  assemble $D_n = \mathrm{diag}(W\mathbf{1}), \quad L_{H_n} = D_n - W$
  $U^{(0)} \leftarrow \mathbf{0}$
  for $t = 0, \dots, T_{\max} - 1$ **do**
    $U^{(t+1)} \leftarrow D_n^{-1}(WU^{(t)} + B_n)$
    if $\|U^{(t+1)} - U^{(t)}\|_F / \max\{1, \|U^{(t)}\|_F\} \leq \varepsilon$ **then**
      break
    end if
  end for
  optionally reweight $U^{(t+1)}$ by class priors
  $\hat{y}_n \leftarrow \arg\max_c U^{(t+1)}(x_n, c)$
end for

3) Linear Convergence. Let U_n^* be the exact Poisson solution on H_n satisfying the gauge condition. Define the spectral radius $\rho_n = \|D_n^{-1} W_n\|_2$. If H_n is connected with at least one supernode of positive degree, then $\rho_n < 1$ and

$$\|U_n^{(t)} - U_n^*\|_F \;\leq\; \rho_n^t \, \|U_n^{(0)} - U_n^*\|_F, \qquad t \geq 0.$$

Moreover, $\rho_n = 1 - \lambda_{\min}^+(L_{H_n})/d_{\max}$, where $\lambda_{\min}^+(L_{H_n})$ is the smallest positive eigenvalue of L_{H_n} and $d_{\max}$ is the maximum weighted degree in H_n.

The connection to a supernode is essential: without it, L_{H_n} has a one-dimensional kernel and the iteration would oscillate. The supernode acts as a ground node (voltage source in the electric analogy), making the system strictly contractive.

4) Error Accumulation Bound. Suppose that at each star–mesh step, the computed reduced Laplacian $\widehat{L}_{\mathrm{red}}$ satisfies $\|\widehat{L}_{\mathrm{red}} - L_{\mathrm{red}}\|_F \leq \delta$ with $\delta = O(\varepsilon_{\mathrm{mach}} \|L\|_F)$. After k sequential reductions, the deviation of the computed Poisson solution $\widehat{U}_T$ from the exact restricted solution U_T^* satisfies

$$\|\widehat{U}_T - U_T^*\|_F \;\leq\; k\, \delta\, \kappa(L_{\mathrm{red}})\, \|U_T^*\|_F \;+\; O(\varepsilon_{\mathrm{mach}}),$$

where $\kappa(L_{\mathrm{red}}) = \lambda_{\max}^+(L_{\mathrm{red}})/\lambda_{\min}^+(L_{\mathrm{red}})$ is the condition number of the reduced Laplacian restricted to its non-null part.

In practice, $k \leq \tau$ (the active window size), $\varepsilon_{\mathrm{mach}} \approx 10^{-15}$ for double precision, and $\kappa(L_{\mathrm{red}})$ is moderate for well-connected local graphs. The bound suggests that numerical issues arise only when the graph becomes nearly disconnected, which can be detected by monitoring $\lambda_{\min}^+$.

5) Drift Stability. Suppose that at timestep n, the edge weights are perturbed by ΔW with $\|\Delta W\|_F \leq \epsilon$. Then the corresponding change in the Poisson solution satisfies

$$\|U_n^{\mathrm{new}} - U_n^{\mathrm{old}}\|_F \;\leq\; \epsilon\, \kappa(L_{H_n})\, \left\|L_{H_n}^+\right\|_2 \|B_n\|_F \;+\; O(\epsilon^2),$$

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1032 DOI: 10.66693/mathai.1032
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

5.2 Computational Complexity

Table 1 summarizes the per-step computational cost and storage of SSLPLP compared with a batch recomputation baseline.

Table 1: Complexity per streaming step. τ: active window; C: classes; m_n: labeled count; T_{iter}: solver iterations; n: stream length; ω: similarity ratio; χ: domain size.

Operation	SSLPLP	Batch recomputation
Edge insertion	$O(\tau + Cm_n)$	$O(n + Cm_n)$
Graph update step	$O(\tau^2)$	$O(n^2)$
Solver (per iteration)	$O(\tau^2 C)$	$O(n^2 C)$
Total per step	$O(\tau^2 C T_{\text{iter}} + m_n)$	$\Omega(n\tau^2 C)$
Memory (bits)	$O(\tau^2 \log(n\omega) + (m_n + \tau)\log\chi)$	$\Omega(n\tau \log(n\omega))$

Compared with TLP Wagner et al. (2018), which costs $O(\tau^3 + m_n)$ per step (matrix inversion to solve the harmonic system), SSLPLP uses an iterative solver whose per-iteration cost is $O(\tau^2 C)$. For small C and sparse graphs the constant T_{iter} is typically in the range 50–200; the two methods are comparable in practice.

6 Experiments

We evaluate on two datasets:

- **Two Moons** (synthetic): $n = 2000$ points from two interleaved half-moons with Gaussian noise $\sigma = 0.1$, streamed in random order. Allows controlled ablation of active window size and label rate.
- **INCART-ECG** ECG time-series annotated with heartbeat arrhythmias (atrial vs. ventricular premature contractions). Streamed in natural temporal order with shingling of $N = 50$ consecutive beats.

Baselines.

- **TLP** Wagner et al. (2018): harmonic label propagation with star–mesh reduction.
- **QLP** Valko et al. (2010): online SSL on quantized graphs.
- **LOLP**: labels-only baseline, ignores all unlabeled data.

All methods are given the same active budget τ and the same labeled data.

We use the RBF similarity $\text{Sim}(x, y) = \exp(-\|x - y\|^2/\sigma^2)$ throughout. The bandwidth σ is set by the median heuristic on a held-out window of 200 points. For temporally ordered streams, points are presented in natural order. The iterative solver uses $T_{\max} = 200$ and $\varepsilon = 10^{-5}$. We report accuracy (proportion of correctly classified unlabeled stream points after an initial burn-in of τ steps), sensitivity, and specificity on binary datasets. All results are averaged over 5 independent runs with different random seeds.

Table 2 presents the main results. Key observations are:

Table 2: Classification accuracy (%) and average processing time per point (ms). All methods use the same labeled data and active budget τ. Results for INCART-ECG averaged over patients.

Dataset	τ	SSLPLP	TLP	QLP	LOLP
Two Moons (2 labels/class)	30	**92.4**	87.1	71.8	63.2
INCART-ECG (2 labels/class)	5	**95.7**	94.9	58.9	70.0

(1) Poisson formulation outperforms harmonic in low-label regime. On Two Moons with 2 labels per class, SSLPLP (92.4%) outperforms TLP (87.1%). This is consistent with the theoretical

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1032 DOI: 10.66693/mathai.1032
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

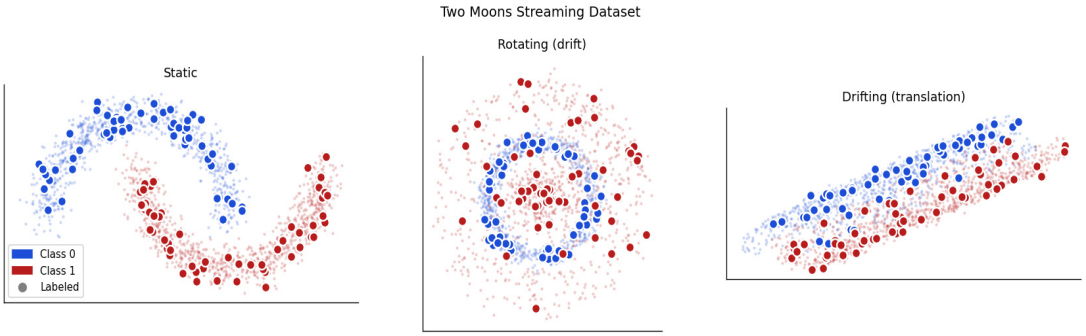


Figure 1: Example of a data-driven Two Moons stream

analysis: the harmonic extension degenerates at very low label rates Nadler et al. (2009), while the Poisson formulation injects sources and sinks that remain informative regardless of label density.

(2) Very few labels suffice. On ECG, 95.7% accuracy with only 2 labels per class demonstrates the practical utility of the Poisson source formulation in extremely sparse-label regimes.

Ablation: Effect of Active Window τ. A representative plot shows classification accuracy on Two Moons as τ varies from 5 to 80, with 2 labels per class. As with TLP Wagner et al. (2018), the relationship between τ and accuracy is non-monotone. Small τ provides insufficient neighborhood context; large τ dilutes the temporal structure and increases computation. An optimal τ around 25–35 balances these effects.

Ablation: Effect of Concept Drift Decay Factor. On a non-stationary variant of Two Moons where the class boundary rotates over time, introducing the decay factor $\alpha \in \{0.95, 0.99, 1.0\}$ in the star–mesh formula equation 2 improves accuracy by 3–8 percentage points over $\alpha = 1$ (no decay), confirming the practical value of the drift-adaptation mechanism of Section 4.6.

Convergence of Iterative Solver. A representative convergence plot shows $\|U^{(t)} - U^*\|_F$ versus iteration t for SSLPLP on the ECG dataset at representative timesteps. The linear convergence predicted by is clearly visible; the empirical rate $\rho \approx 0.82$ matches the spectral bound $1 - \lambda^+_{\min}(L_{H_n})/d_{\max}$.

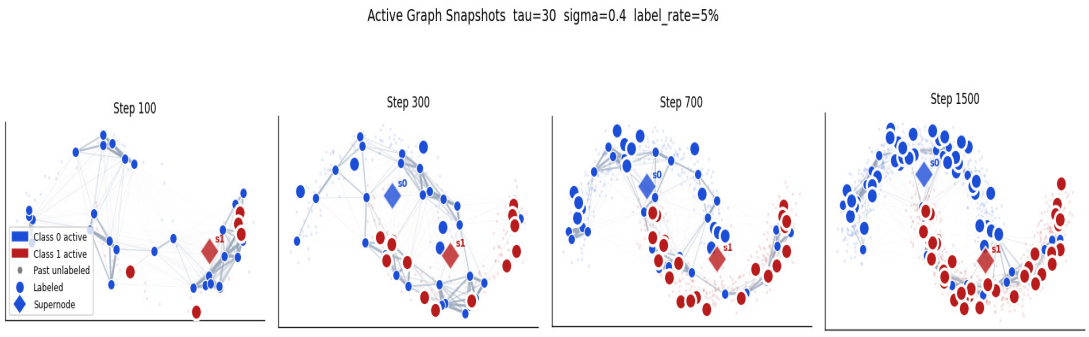


Figure 2: SSLPLP predictions on a data-driven Two Moons stream

6.1 Limitations

Temporal structure dependence. Like TLP Wagner et al. (2018), SSLPLP is primarily designed for streams with local temporal structure. On datasets without natural temporal ordering, it does not

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1032 DOI: 10.66693/mathai.1032
Special Issue: MathAI 2026 Selected Papers
Received: 07 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

outperform simpler alternatives. For unordered data, a kNN-based active neighborhood can be used, but this requires an approximate nearest-neighbor index and introduces additional approximation errors.

Parameter sensitivity. The active window size τ is dataset-dependent and lacks an automatic selection rule. In practice, τ should be set to roughly capture the temporal context span of the data, but this requires domain knowledge or cross-validation. The convergence rate ρ_n and the error bound depend on $\kappa(L_{H_n})$, which increases when the active graph becomes sparse or nearly disconnected. Adding a small regularization term λI to the Laplacian mitigates this but introduces bias.

Computational cost of supernode updates. When a new labeled point arrives, updating $w(s_c, v)$ for all $v \in Q_n$ costs $O(\tau)$ similarity evaluations. For datasets with high-dimensional features and a large active set, this can be the computational bottleneck. Approximate similarity via locality-sensitive hashing could reduce the practical cost of these evaluations.

Non-stationary label distributions. If the empirical label distribution $\bar{y}^{(n)}$ changes rapidly (severe concept drift in the label space), the source term B_n may become ill-conditioned. The sliding supernode mechanism of Section 4.6 partially addresses this, but a full theoretical analysis of the non-stationary case is left for future work.

7 Conclusion

We proposed Semi-Supervised Local Poisson Label Propagation (SSLPLP), a local graph-based framework for semi-supervised classification on dynamic data streams. The method combines a Poisson source–sink formulation, class supernode aggregation, decayed star–mesh graph reduction, and an iterative local Poisson solver.

The main theoretical contributions are: (i) an exact equivalence theorem showing that Schur complement reduction preserves Poisson potentials on the active region ; (ii) a linear convergence theorem for the iterative solver with an explicit spectral rate ; (iii) a bound on the numerical error accumulated over k sequential reductions ; and (iv) a first-order drift stability result .

Experimentally, SSLPLP matches or exceeds TLP on temporally ordered streams and outperforms TLP in the extreme low-label regime, consistent with the known advantage of the Poisson formulation over harmonic extension when labels are scarce. The star–mesh reduction remains an effective mechanism for preserving local temporal information.

References

Konstantin Avrachenkov, Pavel Chebotarev, and Alexey Mishenin, Avrachenkov, Konstantin and Chebotarev, Pavel and Mishenin, Alexey (2017). Semi-supervised learning with regularized laplacian. *Optimization Methods and Software*, 32(2):222–236.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani, Belkin, Mikhail and Niyogi, Partha and Sindhwani, Vikas (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434. URL `https://www.jmlr.org/papers/v7/belkin06a.html`.

Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux, Bengio, Yoshua and Delalleau, Olivier and Le Roux, Nicolas (2006). Label propagation and quadratic criterion. pp., 193–216. MIT Press, Cambridge, MA.

Jeff Calder, Calder, Jeff (2019). The game theoretic p-laplacian and semi-supervised learning with few labels. *Nonlinearity*, 32(1):301–331.

Jeff Calder, Brendan Cook, Matthew Thorpe, and Dejan Slepcev, Calder, Jeff and Cook, Brendan and Thorpe, Matthew and Slepcev, Dejan (2020). Poisson learning: Graph based semi-supervised learning at very low label rates. pp., 1306–1316, PMLR. URL `https://proceedings.mlr.press/v119/calder20a.html`.

Received: 07 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Jeff Calder and Dejan Slepčev, Calder, Jeff and Slepčev, Dejan (2020). Properly-weighted graph laplacian for semi-supervised learning. *Applied Mathematics and Optimization*, 82(3):1111–1159.

Ahmed El Alaoui, Xiuyuan Cheng, Aaditya Ramdas, Martin J. Wainwright, and Michael I. Jordan, El Alaoui, Ahmed and Cheng, Xiuyuan and Ramdas, Aaditya and Wainwright, Martin J. and Jordan, Michael I. (2016). Asymptotic behavior of ℓ_p-based laplacian regularization in semi-supervised learning. pp., 879–906, PMLR. URL https://proceedings.mlr.press/v49/elalaoui16.html.

Boaz Nadler, Nathan Srebro, and Xueyuan Zhou, Nadler, Boaz and Srebro, Nathan and Zhou, Xueyuan (2009). Semi-supervised learning with the graph laplacian: The limit of infinite unlabelled data. pp., 1330–1338. URL https://www.weizmann.ac.il/math/Nadler/sites/math.Nadler/files/publications/semi-supervised_learning_with_the_graph-laplacian-_the_limit_of_infinite_unlabelled_data.pdf.

Sujith Ravi and Qiming Diao, Ravi, Sujith and Diao, Qiming (2016). Large scale distributed semi-supervised learning using streaming approximation. pp., 519–528, Cadiz, Spain, PMLR. URL https://proceedings.mlr.press/v51/ravi16.html.

Dravyansh Sharma and Maxwell Jones, Sharma, Dravyansh and Jones, Maxwell (2023). Efficiently learning the graph for semi-supervised learning. pp., 1900–1910, PMLR. URL https://proceedings.mlr.press/v216/sharma23a.html.

Dejan Slepčev and Matthew Thorpe, Slepčev, Dejan and Thorpe, Matthew (2019). Analysis of p-laplacian regularization in semi-supervised learning. *SIAM Journal on Mathematical Analysis*, 51(3):2085–2120.

Michal Valko, Branislav Kveton, Ling Huang, and Daniel Ting, Valko, Michal and Kveton, Branislav and Huang, Ling and Ting, Daniel (2010). Online semi-supervised learning on quantized graphs. pp., 605–612, AUAI Press. URL https://chercheurs.lille.inria.fr/~valko/hp/publications/valko2010online.pdf.

Krishnamurthy Viswanathan, Sushant Sachdeva, Andrew Tomkins, and Sujith Ravi, Viswanathan, Krishnamurthy and Sachdeva, Sushant and Tomkins, Andrew and Ravi, Sujith (2019). Improved semi-supervised learning with multiple graphs. pp., 3032–3041, PMLR. URL https://proceedings.mlr.press/v89/viswanathan19a.html.

Tal Wagner, Sudipto Guha, Shiva Kasiviswanathan, and Nina Mishra, Wagner, Tal and Guha, Sudipto and Kasiviswanathan, Shiva and Mishra, Nina (2018). Semi-supervised learning on data streams via temporal label propagation. pp., 5095–5104, PMLR. URL https://proceedings.mlr.press/v80/wagner18a.html.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf, Zhou, Dengyong and Bousquet, Olivier and Lal, Thomas Navin and Weston, Jason and Schölkopf, Bernhard (2004). Learning with local and global consistency. pp., 321–328. URL https://proceedings.neurips.cc/paper/2506-learning-with-local-and-global-consistency.pdf.

Xiaojin Zhu and Zoubin Ghahramani, Zhu, Xiaojin and Ghahramani, Zoubin (2002). Technical Report. CMU-CALD-02-107, Carnegie Mellon University. URL https://mlg.eng.cam.ac.uk/zoubin/papers/CMU-CALD-02-107.pdf.

Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty, Zhu, Xiaojin and Ghahramani, Zoubin and Lafferty, John D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. pp., 912–919. URL https://mlg.eng.cam.ac.uk/pub/pdf/ZhuGhaLaf03a.pdf.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1036 DOI: 10.66693/mathai.1036
Special Issue: MathAI 2026 Selected Papers
Received: 09 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

ALIGNING THE NUMBER OF PARAMETERS WITH THE NUMBER OF LINEAR REGIONS FOR IMPROVED NEURAL NETWORK APPROXIMATION

Alexey Podoprosvetov[*], **Vladimir Smolin**[†], **Sergey Sokolov**[‡]
Keldysh Institute of Applied Mathematics,
Miusskaya sq., 4, Moscow, 125047, Russia
llecxis@gmail.com, smolin@keldysh.ru, sokolsm@list.ru

ABSTRACT

The paper addresses the ”black box” problem of neural networks by analyzing the approximation properties of latent layers. It proposes that a key limitation preventing the practical achievement of universal approximation theorems is the mismatch between the growth rates of a network’s parameters and the number of linear regions partitioning the input space. The question is examined how this imbalance is exacerbated in multidimensional cases, hindering effective learning. To resolve this, methods are suggested to align parameter counts with the number of linear regions, such as moving activations vectors to the surface of a hypercube, utilizing micro-columns, and leveraging the ”blessing of dimensionality” in deep networks to decouple complex signals.

1 INTRODUCTION: WHY NEURAL NETWORK APPROXIMATION MECHANISMS REMAIN POORLY UNDERSTOOD

A paradox persists: despite an abundance of textbooks, the inner workings of neural networks are still declared “incomprehensible” Barez et al. (2025). The root cause lies in the impossibility of visually interpreting transformations carried out in high-dimensional spaces. This is especially true of hidden layers: their activity is corrected by backpropagation, yet has no transparent meaning.

As a result, network optimization remains largely empirical, and numerous theories explain learning algorithms rather than the nature of the approximations that have already been formed.

We propose an approach based on topological analysis of the state spaces of latent-layer activation vectors. We examine one of the principal reasons for the discrepancy between practical training and the universal approximation theorem Cybenko (1989): the different growth rates of the number of network parameters and the number of linear regions partitioning the input signal space.

2 SURVEY OF RECENT THEORIES OF NEURAL NETWORK APPROXIMATION

Research into the organisation of knowledge in latent space is actively ongoing. Several directions can be identified:

(a) Application of geometric algebra enables formal investigation of spaces of arbitrary dimensionality without requiring their visual representation Sivgin et al. (2025).

(b) Study of sparse autoencoders (SAE) decomposes complex computations into thousands of human-interpretable concepts Transformer Circuits Thread (2025).

(c) Investigation of the “grokking” phenomenon — a sudden transition of the network from high error to accurate generalisation Power et al. (2022).

[*]ORCID: 0000-0002-3608-7895
[†]ORCID: 0000-0001-9030-6545
[‡]ORCID: 0000-0001-6923-2510

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1036 DOI: 10.66693/mathai.1036
Special Issue: MathAI 2026 Selected Papers
Received: 09 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

(d) Analysis of multi-layer interactions instead of individual neuron contributions He et al. (2024).

(e) Generalisation theory explaining why the found transformations succeed Dunefsky et al. (2024).

Each approach clarifies certain aspects of processes in the latent layers, but none describes their approximation properties as a whole. We propose complementing these methods with topological analysis of activation-vector state spaces, which allows broader conclusions about the nature of approximation.

3 Variables in the Basic Formal-Neuron Network Description

Formal-neuron models, tracing back to McCulloch and Pitts McCulloch & Pitts (1943) and generalised by Widrow and Hoff Widrow & Hoff (1960), include the weight vector $\vec{W}_i^l = \{w_{ji}^l\}$, the *linear* activation a_i^l, and the nonlinear activation o_i^l. The weight vectors form a matrix $W^l = \{\vec{W}_i^l\}$ multiplied by the output of the preceding layer $\vec{O}^{l-1}$ (Fig. 1a):

$$\vec{A}^l = W^l \vec{O}^{l-1}; \quad \vec{A}^l = \{a_i^l = \vec{W}_i^l \, \vec{O}^{l-1}\}; \quad \vec{O}^l = \{o_i^l = \sigma(a_i^l)\}. \tag{1}$$

The nonlinear function $\sigma(x)$ can take various forms (Sigmoid, ReLU, SiLU, etc. Widrow & Hoff (1960)), including negative values and non-monotone dependencies.

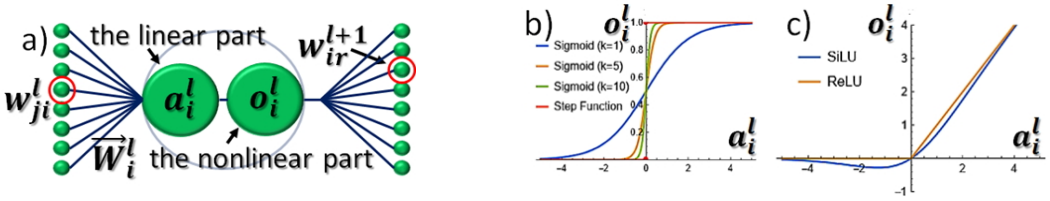


Figure 1: (a) Structure of formal neuron i in layer l: input-connection weight vector $\vec{W}_i^l = \{w_{ji}^l\}$, linear activation a_i^l, and nonlinear activation o_i^l. (b) Nonlinear functions $o_i^l = \sigma(a_i^l)$: Sigmoid, $\sigma_{\text{Sigm}}(x) = 1/(1+e^{-kx})$; (c) SiLU, $\sigma_{\text{SiLU}}(x) = x\,\sigma_{\text{Sigm}}(x)$, and ReLU, $\sigma_{\text{ReLU}}(x) = \max(0, x)$ Hammad (2024).

Network parameters $\vec{\theta} = \{w_{ji}^l\}$ are tuned by *gradient descent to minimise the loss function* $E = \sum_{q=1}^{M} E_q(\vec{\theta})$:

$$\Delta\vec{\theta}_q = -\alpha \sum_{q=1}^{M} \nabla E_q(\vec{\theta}); \qquad \nabla E_q(\vec{\theta}) = \left\{\frac{\partial E_q}{\partial w_{ji}^l}\right\}. \tag{2}$$

Computing the partial derivatives $\frac{\partial E_q}{\partial w_{ji}^l}$ requires knowledge of $\frac{\partial E_q}{\partial a_i^l}$ for each layer. For the output layer these derivatives are determined directly by the approximation error; for hidden layers they are obtained via the backpropagation algorithm Rumelhart et al. (1986):

$$\delta_{iq}^l = \frac{\partial E_q}{\partial a_{iq}^l} = \frac{\partial E_q}{\partial o_{iq}^l}\frac{\partial o_{iq}^l}{\partial a_{iq}^l} = \Big(\sum_{r=1}^{N} w_{ir}^{l+1}\,\delta_{rq}^{l+1}\Big)\sigma'(a_{iq}^l); \qquad \frac{\partial E_q}{\partial w_{ji}^l} = \frac{\partial E}{\partial a_{iq}^l}\frac{\partial a_{iq}^l}{\partial w_{ji}^l} = \delta_{iq}^l\, o_{jq}^{l-1}, \tag{3}$$

where $\delta_{rq}^{l+1} = \frac{\partial E_q}{\partial a_{rq}^{l+1}}$, $\sigma'(a_{iq}^l) = \frac{d\sigma(a_{iq}^l)}{da_{iq}^l}$, and the index q indicates dependence on the specific training pair $(\vec{X}_q, \vec{Y}_q^g)$. Individual values δ_{iq}^l (and correspondingly individual increments $\frac{\partial E_q}{\partial w_{ji}^l}$ from Eq. equation 2) are computed for each pair. Changes in the parameter vector $\Delta\vec{\theta}_q$ according to

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1036 DOI: 10.66693/mathai.1036
Special Issue: MathAI 2026 Selected Papers
Received: 09 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Eq. equation 2 occur after a relatively randomly chosen subset (M pairs) of the training set $(\vec{X}_q, \vec{Y}_q^g)$ is presented.

Pitfalls in deep learning are largely due to two factors. First, the need to reconcile in Eq. equation 2 the parameter increments composed of independently directed gradients ∇E_q for different q. Second, the finiteness of the learning steps. Infinitesimally small increments would ensure linearity of the influence of layer-by-layer changes, but training would take infinitely long. Choosing the largest feasible α, on the other hand, leads to unequal influence of parameter changes at different network depths.

It is shown in Podoprosvetov et al. (2024) that gradient descent can in principle be driven by adjusting only the output layer; yet deep networks demonstrate the ability to solve complex problems. This means that stochastic gradient descent according to Eq. equation 2 is capable of supporting diverse optimisation methods implemented on multi-layer structures. Increasing the number of layers and using various regularisation methods raises the probability of forming successful approximations.

4 Pruning

The flip side of the stochastic nature of neural network model formation is the presence of regions that are poorly tuned during training. Removing them from the structure — *pruning* — does not impair functioning and allows more efficient models to be created Cheng et al. (2024). This is one of the most dynamically developing directions in deep learning, having progressed from empirical methods to rigorous mathematical theories.

Modern methods such as PrunedLoRA propose dynamic elimination of less important components during fine-tuning, yielding more compact adapters compared with the standard LoRA technique Yu et al. (2025).

Applying pruning can reduce computational costs at inference and during the final stages of training by several times (sometimes by an order of magnitude). However, in the early stages considerable effort is expended on tuning parameters that will subsequently be discarded.

5 The Universal Approximation Theorem — Why It Does Not Work in Practice

In 1989, Cybenko proved the universal approximation theorem Cybenko (1989): a network with a single hidden layer and sigmoid nonlinearity can approximate any continuous function to any prescribed accuracy as the number of elements grows. However, the theorem only proved the *existence* of such settings, and gave no constructive path to achieving them. The theorem was subsequently extended to a broad class of nonlinear transformations $\sigma(x)$.

It was shown in Fokina & Oseledets (2019) that even knowing the exact weight values does not guarantee finding them via the standard SGD algorithm. The authors proposed a method of exponential error reduction by identifying regions with the largest error and adding new neurons at those locations. An alternative approach based on neural-network mapping that distributes the nonlinear properties of elements across the input signal space was proposed in Shen & Smolin (2025). Both methods reduce the approximation error for analytically defined functions, but the key property of Cybenko's theorem — the error tending to zero — is reproduced only for functions of a single variable. For multi-dimensional cases, zero error was not achieved in experiments.

The present article describes one of the main reasons for this discrepancy: without special measures or "luck" in the stochastic formation of the approximation, there are insufficient parameters for independently tuning all piecewise-linear regions of the input-space partition. The reason is the differing growth rates of the number of parameters and the number of regions as the number of hidden-layer elements increases.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1036 DOI: 10.66693/mathai.1036
Special Issue: MathAI 2026 Selected Papers
Received: 09 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

6 Matrix-Vector Representation of Activations and Weights and Their Interrelation

Before presenting the main idea, we consider certain properties of transformations in neural network elements. Figure 2 shows the structure and parameter notation of a deep network under a vector-matrix description of its operation.

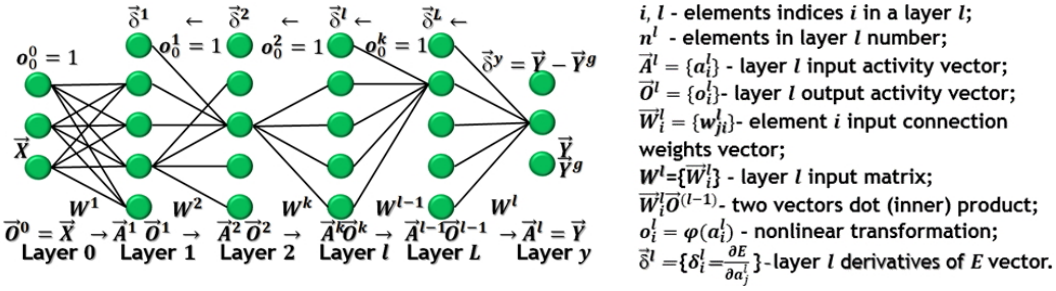


Figure 2: Structure of a deep neural network with L fully connected hidden layers.

The geometric connection between the weight vectors $\vec{W}_i^l$ and the output activity vectors $\vec{O}^{l-1}$ of the preceding layer rests on their shared dimensionality, which allows the linear activation to be computed as a dot product: $a_i^l = \vec{W}_i^l \cdot \vec{O}^{(l-1)}$ (matrix $W^l = \{\vec{W}_i^l\}$, $\vec{A}^l = W^l \vec{O}^{l-1}$). The dimensionality of the "error" vector δ_{iq}^l (Eq. equation 3), which propagates back through the network, likewise coincides with the dimensionality of $\vec{A}^l$, enabling comparison of the changes $\Delta\vec{A}^l$ with the vectors δ_{iq}^l. The coincidence of dimensionalities and the presence of dependencies between vectors (or their increments) of different natures permits a geometric interpretation of certain mutually dependent properties.

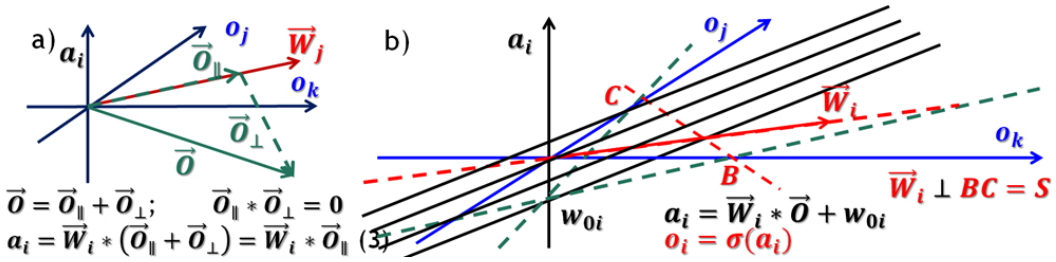


Figure 3: Geometric comparison of vectors $\vec{W}_i$ and $\vec{O}^{l-1}$ (indices l and $l-1$ omitted): (a) the vector $\vec{O}$ can be decomposed into components $\vec{O}_\parallel$ and $\vec{O}_\perp$; (b) partition of the input signal space (output activity $\vec{O}^{l-1}$ of the preceding layer) into two half-spaces by the line BC.

The vector $\vec{O}^{l-1}$ can be expressed as the sum of two components: $\vec{O}_\parallel$ parallel to $\vec{W}_i$, and $\vec{O}_\perp$ perpendicular to it (Fig. 3a). Because the dot product of perpendicular vectors is zero, only the component $\vec{O}_\parallel$ contributes to the linear activation a_i. In a space of dimension n_{l-1} — the number of elements in layer $l-1$ — $(n_{l-1}-1)$ perpendiculars can be drawn to the vector. They form a hyperplane S that divides the space of vectors $\vec{O}^{l-1}$ into two subspaces: on one side of the (hyper)linear surface (in the direction of $\vec{W}_i$), $a_i = \vec{W}_i \cdot \vec{O}_\parallel > 0$; on the other side, $a_i < 0$. On the surface S itself, shown in Fig. 3b as line BC, $a_i = 0$. As the dimensionality n of the space grows, the boundary separating it into two parts also increases its dimensionality, as illustrated in Fig. 4.

7 Non-Locality of Neural Network Approximation

The realisable-state subspace $\hat{O} = \{\vec{O}_q\}$ is the set of all activation vectors that can be obtained when transforming input signals $\vec{X}$ from the approximated domain under fixed parameters $\vec{\theta}$. The

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1036 DOI: 10.66693/mathai.1036
Special Issue: MathAI 2026 Selected Papers
Received: 09 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

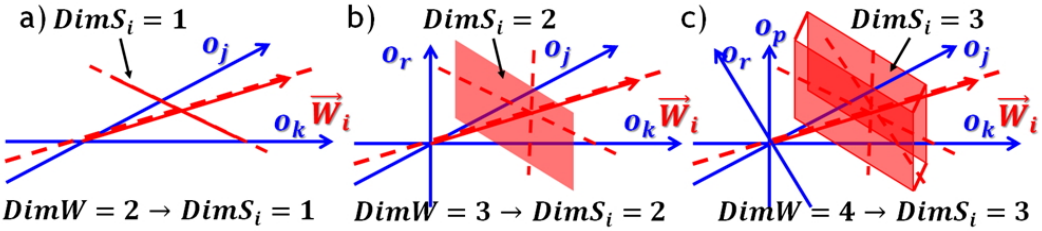


Figure 4: Growth of the dimensionality of subspaces $S_i : a_i = 0$, perpendicular to vectors $\vec{W}_i$, with increasing dimensionality of the spaces $W = \{\vec{W}_i\}$ and $O = \{\vec{O}\}$ (their dimensionalities are equal).

dimensionality of $\hat{O}$ is always less than that of the full space $O = \{\vec{O}\}$, but is generally greater than 1. When ReLU is used, the subspace $\hat{O}$ has a continuous piecewise-linear structure.

Each hyperplane S_i (*the boundary where* $a_i = 0$), intersecting with $\hat{O}$, divides it into two parts. The characteristic sizes of these parts are comparable with the size of the entire subspace $\hat{O}$ and depend only on the orientation of the vectors $\vec{W}_i$. As the number of elements in the layers grows, these sizes do not tend to zero.

Nevertheless, the number of boundaries S_i and the number of their intersections grows. Since the partition regions do not overlap, and the measure of the subspace $\hat{O}$ does not depend on the number of network elements (only the degree of its "crumpling" does), increasing the number of boundaries leads to smaller region sizes. Thus, subdivision of the realisable-state space into progressively finer regions is achieved not by locally reducing the scale of each boundary, but by globally increasing the number of their intersections — this is how the non-local character of neural network approximation manifests itself.

8 One-Dimensional Single-Layer Approximation

To understand multi-dimensional approximation, we first consider the case of a one-dimensional nonlinear function $y^g(x_1)$ (Fig. 5). The network contains one hidden layer; both the input and hidden layers also include an element with constant unit activity to implement biases. Network operation is described by:

$$a_i = \vec{W}_i^1 \vec{X}; \quad o_i = \mathrm{ReLU}(a_i); \quad y = W^2 \vec{O}^1. \tag{4}$$

In Fig. 5 the function is approximated by three linear segments on the interval $[\min(x_1), \max(x_1)]$. Each segment is provided by its own ReLU function; the nonlinear activity of the second and third elements determines the positions of the kink points p_1 and p_2.

The first element has its kink point outside the considered interval, and its parameters w^1_{01} and w^1_{11} are tuned to match the middle segment. The parameters of the second element, w^1_{02} and w^1_{12}, allow the activation point to be set at $x_1 = p_1$ and the slope to be chosen so that the sum with the first element matches the right segment. The third element is analogously tuned for the left segment.

Accuracy can be increased by adding hidden-layer elements, introducing new kink points and progressively improving the approximation. This procedure corresponds to the approach of Fokina & Oseledets (2019) and allows any prescribed error ε to be achieved for any continuous function of one variable, demonstrating the fulfilment of Cybenko's theorem Cybenko (1989) in the one-dimensional case.

It is important to note that as the number of hidden-layer elements grows, not only the number of parameters but also the number of linear approximation segments grows proportionally. This is precisely what ensures the required accuracy. For multi-variable functions, as will be shown, the condition of matching the number of parameters to the number of linear regions may be violated, requiring either a fortuitous configuration during SGD or special adaptive algorithms.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1036 DOI: 10.66693/mathai.1036
Special Issue: MathAI 2026 Selected Papers
Received: 09 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

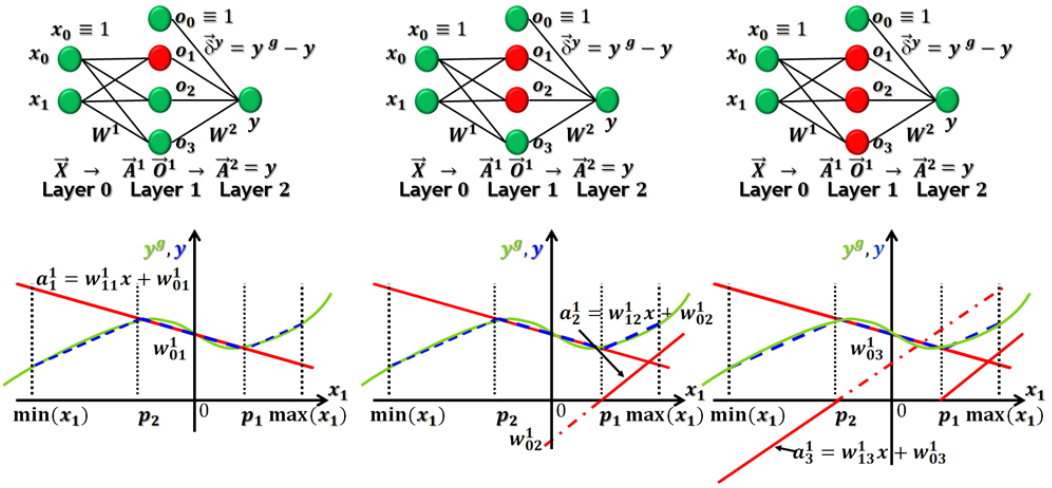


Figure 5: Piecewise-linear approximation of a function of one variable: the target function y^g is approximated by successively activating three hidden-layer elements.

9 Two-Dimensional Single-Layer Approximation

In the one-dimensional case, each additional boundary S_i (point) divides one linear region into two. In the two-dimensional case the boundaries S_i are straight lines of infinite extent, capable of crossing several regions of the subspace $\hat{O}$ simultaneously, creating several new partition regions.

Figure 6 shows the partition of the subspace by two element boundaries. They allow independent tuning of the properties of transformations in only two of the four delineated regions. Adding an element whose boundary does not intersect $\hat{O}$ creates no new regions but adds parameters for tuning. Adding elements with intersecting boundaries increases the number of regions faster than the number of parameters.

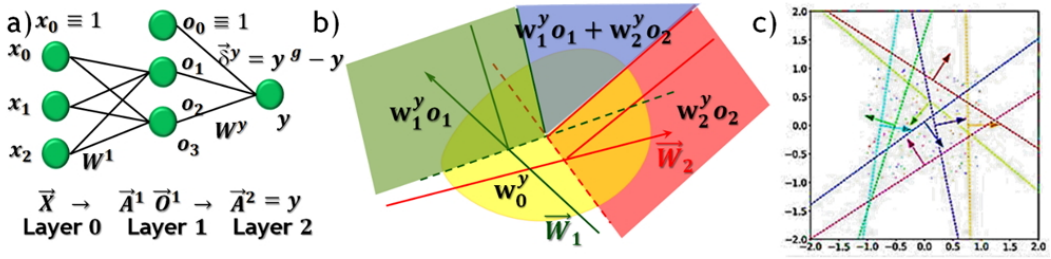


Figure 6: Piecewise-linear approximation of a function of two variables: (a) network structure; (b) piecewise-linear surfaces; (c) partition of the plane by eight boundaries S_i.

As shown in Fig. 7, the growth of the number of regions is quadratic in character (but depends not only on the number of neurons and the input dimensionality). Research Montúfar et al. (2014) shows that the maximum number of linear regions for a single-layer ReLU network, and the parameters required to describe them, can exceed the number of connection weights, which at a fixed parameter count limits the approximation capacity.

In general, when considering the two-dimensional partitioning of the subspace $\hat{O}$ into regions by boundaries S_i, as shown in Fig. 7, the growth of the number of regions is quadratic, but the dependence is not strict: by varying the positions of boundaries S_i the number of regions can be both increased and decreased.

Exact approximation by a single-layer perceptron is achievable only for functions of a special form admitting variable separation:

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1036 DOI: 10.66693/mathai.1036
Special Issue: MathAI 2026 Selected Papers
Received: 09 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

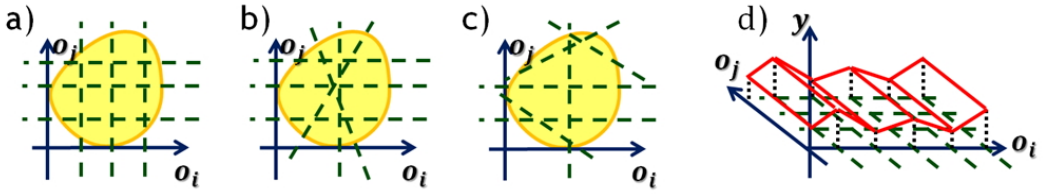


Figure 7: Piecewise-linear approximation of a function of two variables: (a)–(c) variants of partitioning the subspace $\hat{O}$ into regions by boundaries S_i; (d) one-dimensional approximation of a two-dimensional region.

$$y^g(\vec{X}) = \sum_{i=1}^{U} y^g(\mu_i^l) \quad \text{or} \quad y^g(\vec{X}) = \prod_{i=1}^{U} y^g(\mu_i^l), \quad \text{where} \quad \mu_i^l = \sum_{i=1}^{U} \beta_i^l o_i^l. \tag{5}$$

In this case the approximation problem reduces to the one-dimensional case (multiple one-dimensional dependencies, U in total, can be approximated simultaneously and summed at the output layer). Approximation of general-form functions requires multi-layer networks.

10 Neural Network Mapping, Orthogonalisation, and Curvature Control for Approximation Problems

Improving approximation accuracy need not rely solely on an excessive number of elements followed by pruning. An opposite approach is also possible: increasing the probability of achieving training results in which all parameters are used efficiently and pruning is not required. The present article is dedicated precisely to aligning the number of parameters with the number of linear approximation regions.

Other methods for increasing parameter efficiency also exist. Reference Podoprosvetov et al. (2024) describes approaches to distributing parameter participation in approximating different parts of the transformation via neural network mapping. Questions of orthogonalisation and curvature tuning of nonlinear transformations, which reduce the mutual influence of approximation regions, are addressed in Shen & Smolin (2025).

Achieving high approximation accuracy calls for the combined use of all these methods, including regularisation and normalisation, and not only the alignment problem addressed in this article.

11 Two-Dimensional Multi-Layer Approximation

The boundaries S_i are linear with respect to the variables of the preceding layer $\vec{O}^{(l-1)}$. However, due to the nonlinear transformations $o_i^{l-1} = \sigma(a_i^{l-1})$, these boundaries are no longer linear with respect to the input signal $\vec{X}$, especially in deep layers. Computer simulation allows the values of $\vec{X}$ to be identified at which the condition $a_i^l = 0$ holds for elements of different layers (Fig. 8a).

As expected, the shapes of the boundaries S_i with respect to the input domain $\vec{X}$ turn out to be nonlinear; since ReLU was used, the nonlinearity takes the form of piecewise-linear broken curves.

As shown in Fig. 8b, the shape of the boundaries S_i with respect to $\vec{X}$ is piecewise-linear. This is explained by the fact that the linear boundary (hyperplane) intersects the broken subspace $\hat{O}^{l-1}$, generating a one-dimensional broken curve. The kinks arise precisely at those values of $\vec{X}$ where $a_i^l = 0$ held in previous layers, i.e., at the intersections of preceding-level boundaries. Research shows that such cascaded boundary intersections do not lead to exponential growth of the number of linear regions with network depth, preserving a power-law dependence.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1036 DOI: 10.66693/mathai.1036
Special Issue: MathAI 2026 Selected Papers
Received: 09 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

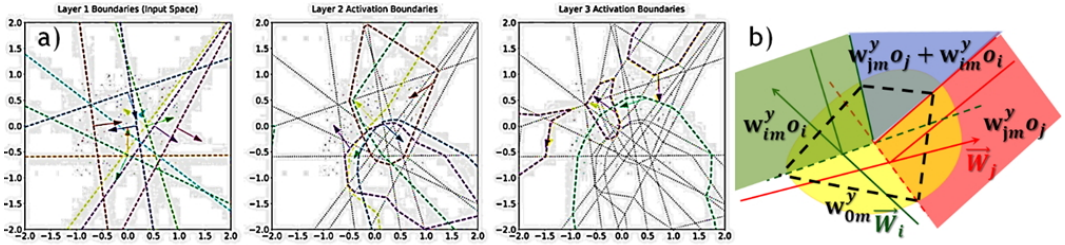


Figure 8: Piecewise-linear approximation of a function of two variables: (a) shapes of the partition of the subspace $\hat{X}$ into regions by boundaries S_i in different layers; (b) intersection of the boundary S_i, linear with respect to the activity of the preceding layer $\vec{O}^{(l-1)}$, with the piecewise-linear shape of the subspace $\hat{O}^{(l-1)}$.

For three or more essential variables, visualisation becomes extremely difficult: the subspaces $\hat{O}^l$ have a three-dimensional broken shape and, intersecting with boundaries S_i, form two-dimensional broken boundaries with respect to $\hat{X}$, making their depiction on a plane largely uninformative.

12 Dimensionalities of Subspaces $\hat{O}^l$ in High-Dimensional Spaces O^l

Since hidden layers may contain hundreds and thousands of elements, the dimensionality of the vectors $\vec{O}^l$, equal to their number of components, can be very large. Why should the realisable-state space $\hat{O}^l$ have lower dimensionality? Consider two cases:

(a) **A small number of input-signal components.** If the number of components of the input signal X is small, then for any values of X small changes in the corresponding vectors $\vec{O}^l$ can only occur in the directions of their increments under differentially small changes in the input signal components or any linear combination thereof:

$$d\vec{O}^l(\vec{X}) = \sum_{i=1}^{n_x} c_i \frac{\partial \vec{O}^l}{\partial x_i}\, dx_i; \qquad \frac{\partial \vec{O}^l}{\partial x_j} = \left\{ \frac{do_i^l}{dx_j}\, e_i^l \right\}. \tag{6}$$

The number of independent directions does not exceed n_x and may become even smaller if certain directions degenerate under nonlinear transformations. Consequently, the dimensionality of the subspaces $\hat{O}^l$ for any layer l cannot exceed the number of input signal components.

(b) **A large number of sensors but few degrees of freedom of the signal source.** If the input signal has many components (thousands or millions) but the observation targets objects with a small number of degrees of freedom, then the vectors $\vec{O}^l \in \hat{O}^l$ can only change in the directions of increments under changes in the variables describing those degrees of freedom. Consequently, regardless of the number of input-signal components, the dimensionality of $\hat{O}^l$ cannot exceed the number of degrees of freedom of the observed objects.

Experimental studies confirm that representations in deep networks are indeed concentrated on manifolds of substantially lower dimensionality than that of the layers themselves Ansuini et al. (2019).

13 Growth of the Number of Parameters with the Number of Network Elements

In multi-layer perceptrons the number of connections N_w grows proportionally to the square of the number of elements N_e:

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1036 DOI: 10.66693/mathai.1036
Special Issue: MathAI 2026 Selected Papers
Received: 09 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

$$N_w = \sum_{l=0}^{L} n^l n^{l+1}; \quad \text{for } n^1 = n^l = n^L: \quad N_w \sim (L-1)\left(\frac{N_e}{L}\right)^2, \quad \Rightarrow \quad N_w(N_e) \sim \frac{N_e^2}{L}. \tag{7}$$

At the same time, the growth of the number of regions when the dimensionality of the subspaces $\hat{O}^l$ exceeds two follows a power law corresponding to that dimensionality.

From estimate equation 7 it follows that the maximum number of parameters (connection weights) is achieved at $L = 2$. Yet it is well known that deep networks with hundreds and thousands of layers are effective for complex problems. What is the reason?

Our view: achieving high accuracy when approximating multi-dimensional transformations with a single-layer perceptron is impossible — the growth of the number of regions catastrophically outpaces the growth of the number of parameters. Deep networks, despite having fewer parameters for the same number of elements, possess properties that allow them to overcome this drawback.

14 The Blessing of Depth and Dimensionality

Bidirectional propagation of data and errors creates conditions for decomposing complex signals and identifying within them the essential variables that influence approximation accuracy. A large number of layers increases the probability of forming such transformations.

High dimensionality of the state vectors $\vec{O}^l \in \hat{O}^l$ creates conditions for forming a large number of mutually perpendicular regions in the subspace $\hat{O}^l$. This allows the shape of the subspace to be gradually changed, reducing the mutual influence of regions during adaptation and slowing the growth rate of their number with increasing N_e.

This phenomenon is known as the "blessing of dimensionality" — in contrast to the "curse of dimensionality" — and is actively studied in the current literature Kůrková (1997).

15 Paths to Aligning the Number of Parameters with the Number of Regions

Since the growth of the number of piecewise-linear approximation regions is proportional to the number of hidden-layer elements raised to a power equal to the dimensionality of the state subspaces $\hat{O}^l$, and generally exceeds the quadratic growth of the number of parameters, achieving alignment requires finding ways to reduce the number of approximation regions. Several methods can be identified:

(a) **"Squeezing" the localisation of subspaces $\hat{O}^l$ onto the surface of the hyperspace O^l.** Normalisation methods (e.g., BatchNorm Ioffe & Szegedy (2015)) compute the mean $\vec{M}^l$ and variance $\vec{D}^l$. When training signals are presented, each component o_i^l is compared with m_i^l and d_i^l, and the loss-function derivatives are corrected by the formulas:

$$\frac{\partial E_q}{\partial o_{iq}^l} = \begin{cases} \alpha(m_i^l - 2d_i^l - o_i^l), & o_i^l < m_i^l, \\ \alpha(m_i^l + 2d_i^l - o_i^l), & o_i^l \geq m_i^l. \end{cases} \tag{8}$$

This keeps activations near the surface of the hypercube, limiting region growth while keeping them away from the central part.

The computed values $\frac{\partial E_q}{\partial o_{iq}^l}$ allow calculation of $\frac{\partial E_q}{\partial a_{iq}^l}$ and $\frac{\partial E_q}{\partial w_{ji}^l}$ via Eq. equation 3, and then parameter increments $\Delta\vec{\theta}_q$ via Eq. equation 2.

Moving the realisable vectors $\vec{O}^l$ towards the hypercube boundaries creates conditions for them to reach vertex regions, where the infinite boundaries S_i may intersect only a small portion of the hypercube containing the subspace $\hat{O}^l$. Fewer other boundaries S_i are intersected, and a smaller number of regions is formed. For high-dimensional space the number

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1036 DOI: 10.66693/mathai.1036
Special Issue: MathAI 2026 Selected Papers
Received: 09 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

of hypercube vertices, equal to 2^l and already exceeding 1000 at $n^l = 10$, is practically unlimited for $n^l > 20$.

The parameter increments for hidden-layer matrices that shift activation vectors $\vec{O}^l$ towards hypercube surfaces can be combined in various proportions with increments computed by backpropagation, regularisation terms, and other parameter optimisation methods.

(b) Alignment can be achieved not only by reducing the number of piecewise-linear approximation regions, but also by increasing the number of parameters per network element. Nonlinear transformations $\sigma(a_i^l)$ can themselves have tunable parameters. Replacing ReLU with VReLU (V-shaped Rectified Linear Unit) with two independent ray-slope parameters slightly increases the total number of network parameters.

An effective approach is the formation of "micro-columns" — small groups of elements in each layer sharing a single output. The output of a "micro-column" is formed on the principle $\max_i a_i^l$ among group members. This approach not only multiplies the number of network parameters several times, but also allows localisation of the boundaries S_i of the elements constituting the micro-column.

(c) Orthogonalisation of the arrangement of regions of the state subspace of the hidden-layer input signal $\hat{O}^l$ in the high-dimensional hyperspace O^l, discussed in Shen & Smolin (2025), has also been confirmed by experiments to reduce the excess number of piecewise-linear approximation regions.

(d) Decomposition of the transformation $\vec{X} \to \vec{Y}$ into components can yield the greatest effect by reducing the dimensionality of the components relative to the original transformation. However, this is a very broad topic and a detailed treatment of its many aspects lies beyond the scope of the present article.

16 Conclusions

This article has examined the problems of aligning the number of network parameters N_w with the number of piecewise-linear approximation regions. The analysis shows that for successful neural network transformations, a decomposition of complex functions into low-dimensional components is necessary — this is an important path towards higher approximation accuracy.

The ReLU case considered here generalises readily to most other nonlinear functions, since practically all of them tend to linear asymptotes as the modulus of the argument grows Hammad (2024).

The importance of signal decomposition is confirmed by the successes of LLM and MLLM architectures OpenAI (2023), which exploit the decomposition of descriptions of a complex world into simple objects and actions already present in texts. However, a discussion of the problems of decomposing complex signals is a topic for a separate article.

References

Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

F. Barez, T.-Y. Wu, I. Arcuschin, et al. Chain-of-thought is not explainability. *Oxford Martin AI Governance Initiative*, 2025. University of Oxford, 15 July 2025. Available at: `https://aigi.ox.ac.uk/publications/chain-of-thought-is-not-explainability/`.

Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10558–10578, 2024.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989. doi: 10.1007/BF02551274.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature circuits. *Advances in Neural Information Processing Systems*, 37:24375–24410, 2024.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1036 DOI: 10.66693/mathai.1036
Special Issue: MathAI 2026 Selected Papers
Received: 09 May 2026 · Accepted: 15 May 2026 · Published: 22 May 2026

Daria Fokina and Ivan Oseledets. Growing axons: greedy learning of neural networks with application to function approximation. *arXiv preprint arXiv:1910.12686*, 2019.

MM Hammad. Deep learning activation functions: Fixed-shape, parametric, adaptive, stochastic, miscellaneous, non-standard, ensemble. *arXiv preprint arXiv:2407.11090*, 2024.

Zhonghao He, Jascha Achterberg, Katie Collins, Kevin Nejad, Danyal Akarca, Yinzhu Yang, Wes Gurnee, Ilia Sucholutsky, Yuhan Tang, Rebeca Ianov, et al. Multilevel interpretability of artificial neural networks: leveraging framework and methods from neuroscience. *arXiv preprint arXiv:2408.12664*, 2024.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. pmlr, 2015.

Věra Kůrková. Dimension-independent rates of approximation by neural networks. In *Computer Intensive Methods in Control and Signal Processing: The Curse of Dimensionality*, pp. 261–270. Springer, 1997.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.

Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *Advances in neural information processing systems*, 27, 2014.

OpenAI. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

Alexey Podoprosvetov, Vladimir Smolin, and Sergey Sokolov. Possibilities for increasing the accuracy of neural network approximation of nonlinear functions for control problems. In *International Conference on Intelligent Systems*, pp. 45–58. Springer, 2024.

A. Power, V. Goel, et al. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

H Shen and VS Smolin. Deep mapping algorithm for more effective neural network training. *Optical Memory and Neural Networks*, 34(Suppl 1):S83–S93, 2025.

I. Sivgin, S. Fridovich-Keil, G. Wetzstein, and M. Pilanci. Geometric algebra planes: Convex implicit neural volumes. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, 2025. Available at: `https://icml.cc/virtual/2025/poster/43546`.

Transformer Circuits Thread. Circuit tracing: Revealing computational graphs in language models. Anthropic, 2025. Available at: `https://transformer-circuits.pub/2025/attribution-graphs/methods.html`.

B. Widrow and M. E. Hoff. Adaptive switching circuits. Technical report, Stanford Electronics Laboratories, Stanford University, 1960. IRE WESCON Convention Record, Part 4, pp. 96–104.

Xin Yu, Cong Xie, Ziyu Zhao, Tiantian Fan, Lingzhou Xue, and Zhi Zhang. Prunedlora: Robust gradient-based structured pruning for low-rank adaptation in fine-tuning. *arXiv preprint arXiv:2510.00192*, 2025.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

TASK-BASED ENGINEERING

Dmitry I. Sviridenko[1]

[1] *Artificial Intelligence Research Center, Novosibirsk State University, Novosibirsk, Russia*

dsviridenko47@gmail.com

ABSTRACT

The article describes the main methodological principles of a new scientific-engineering discipline — task-based engineering, which represents a synergistic integration of the task-based approach, engineering and agent-based AI. The features of each component of task-based engineering are examined, including their advantages and disadvantages. A conceptual model of the new discipline is presented, and its key principles, methodological framework and application potential are elucidated. Differences from related fields are highlighted, and the challenges and development prospects are outlined.

Keywords: task-based engineering, weakly structured need, task-based approach, domain semantic model, solution criterion, contextual conditions, task formalization, semantic modelling, engineering, prompt engineering, context engineering, agent-based AI, AI agents, multi-agent systems

The foundation of any intellectual activity is the ability to formulate and solve problems. However, a serious obstacle in this area stems from two key factors: on the one hand, the lack of a discipline that explains and facilitates an effective transition from an initial weakly structured need to a clearly formulated task; and on the other hand, the complexity and multi-component nature of the resulting well-defined problems — including their hierarchical structure, multi-criteria nature, dynamic environment, data scarcity, the need to account for heterogeneous factors, etc.

Traditional methods for problem formulation and solving often prove insufficient due to:

- vagueness in the initial need's formulation;
- the non-trivial nature of the process of transitioning from a weakly structured problem (expressed as a need) to a precise task formulation;
- neglect of contextual dependencies;
- linear planning of actions for transitioning from a weakly structured problem to a well-defined task and subsequently solving it, without adaptation to possible changes in conditions.

Naturally, this gives rise to the need for a methodology, theory, and technology that would take the above factors into account and help overcome these shortcomings.

In the 1980s, a methodology called the **task-based approach** [1–8] was proposed and has continued to develop actively, along with its logical-mathematical theory — **semantic modelling** [9–24, 39–45]. According to the task-based approach, when discussing a task, we are required to present its following components: a semantic model of the domain to which the task belongs; query(ies) directed to the semantic model, the answer to which constitutes the task's solution; a solution criterion — i.e., whether what is presented as the answer to the query actually qualifies as a solution; contextual conditions. Practice has confirmed the promise and effectiveness of the task-based approach and its mathematical theory — semantic modelling. However, it is worth noting that the methods of the task-based approach and the algorithms of semantic modelling developed to date pay little attention to the process of transforming a weakly structured problem into a well-defined task. Furthermore, the language and algorithms of semantic modelling are primarily oriented towards static task formulation and the problem-solving process, failing to provide adaptation of task formulation and action planning to changing conditions.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

During the same period, there was a rapid emergence and development of the methodology, theory, methods, algorithms, and systems of **machine learning** — in particular, **large language models** and **multimodal models** (LLMs, LMMs). The advent of these models marked new, unprecedented opportunities in the computer "understanding" of natural-language queries and in answer generation. Nevertheless, AI systems based on neural networks often exhibit unpredictable behaviour with respect to specified requirements and context, vulnerability to attacks via subtle distortions of input data, an inability to consistently account for context, a combination of risks and opportunities, and limited capacity to effectively leverage expert knowledge. When using modern generative AI models, mechanisms for interacting with them become particularly important. One of the key interaction mechanisms is the so-called **prompt engineering** (see, for example, Google's excellent guide to prompt engineering [26]). It quickly became clear that the performance and effectiveness of generative models are largely determined not only by the adequacy of query formulation but also by the context they receive. Over time, this context — ranging from simple instructions to guidance on using complex external knowledge bases — gradually became another primary mechanism for controlling the behaviour of a generative model, expanding its knowledge, and unlocking its capabilities. Two formal disciplines have gradually emerged: **prompt engineering** and **context engineering** [26–29]. From the author's perspective, these two disciplines — which mutually complement and enrich each other — should be considered as a single discipline: **engineering**. This discipline, like the task-based approach, is oriented towards automating the processes of problem formulation and solution search. However, unlike the task-based approach, this engineering can only act as a passive toolkit.

In addition to generative artificial intelligence, another traditional area in AI has recently become extremely popular again: **agent-based AI** — a field concerned with the creation of **AI-agents** and **multi-agent systems** (MAS) (here, we can recommend the excellent three-volume monograph [25]). Today, many generative AI-models, as well as a number of specialized frameworks and platforms, enable the automatic creation of agents with various roles and competencies. They also provide the ability to integrate AI-agents into specialized communities — i.e., multi-agent systems. It should be noted that modern agent-based AI distinguishes three main directions: **Software agents** (**program agents**), particularly **LMM-agents**, which operate in the digital world, **Embodied agents**, including **cyber-physical agents**, which possess "bodies" and interact with the real world, **Hybrid agents**, which represent a combination of software and embodied agents.

Since all the aforementioned areas are, to varying degrees, oriented towards problem solving, an analysis of their current state and capabilities naturally led the author to the desire to unify all three disciplines. Their synergistic integration constitutes **task-based engineering**. In fact, this article discusses the emergence of a new scientific and engineering discipline that unites:

1. **The task-based approach** — the *core* of task-based engineering, its system-forming framework, which sets the direction for transforming a weakly structured problem into a well-defined task and subsequently solving it.
2. **Engineering**, represented by:
   - **Prompt engineering** — a *tool* that enables dialogue with a generative AI-model (e.g., an LLM/LMM) for efficient information retrieval, content generation, and refinement of task components;
   - **Context engineering** — a *tool* through which one can create a "working environment" for the need-satisfaction process by managing its surroundings. In the case of using LLM/LMMs, this includes the system prompt, meta-instructions, retrieval of relevant data, dialogue history, etc.
3. **Agent-based AI**, including **AI-agents** and **multi-agent systems (MAS)**, which enables the organization of an effective process for transitioning from a weakly structured problem to a well-defined task, and for subsequently solving complex, multi-component problems.

As noted above, the core of this new discipline is the **task-based approach**, which addresses the following questions: what a task is as an entity, where and how it emerges, how it should be formulated and solved, what types of tasks exist, etc. It should be noted that it was Academician A.N.Kolmogorov who first spoke of tasks as entities deserving separate attention and thorough study — in his early work [30], published back in 1932. There, while describing the denotational semantics of intuitionistic propositional calculus in terms of tasks, he effectively laid out the **axiomatics of**

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

task calculus for the first time. Interestingly, at the time this article was written, it was still unclear what should be understood by a "general method for solving" tasks of a certain class, and what it means to "reduce the solution of one task to the solution of another" [31].

The formal study of tasks, initiated in the 1930s by A.N.Kolmogorov, was independently continued by researchers in other disciplines. Among them, in the author's view, a notable place is occupied by the **Theory of Inventive Problem Solving (TRIZ)** [32–35], proposed in the 1950s by the Soviet patent engineer G. Altshuller. Here, the task-based approach was formulated as a purely empirical discipline, primarily oriented towards the practical, engineering application of its principles. According to TRIZ, a **task** is essentially nothing more than the need to overcome a contradiction between the desired outcome and the existing capabilities of the Problem Solver at each stage of the synthesis/analysis of a Technical System (TS).

The task-based approach received further significant development in the works of Academician Yu.L.Ershov and D.Sci. (Phil.) K.F.Samokhvalov, which are devoted to applying the ideas and principles of the task-based approach to the foundations of mathematics [36, 37]. A key point of the proposed approach is the conclusion that knowing a need "...does not mean knowing the desired outcome, but only means having the ability to recognize the desired outcome" — or, in other words, possessing a **criterion for need satisfaction.** Thus, a task — as a certain need — is defined (conceptualized) if and only if there exists a **solution criterion**: a means of verifying whether the presented construct actually constitutes a solution to the task. Otherwise, any proposed consideration could be considered a solution to the task.

The propositions formulated and substantiated in the works of Yu.L.Ershov and K.F. Samokhvalov were subsequently developed by a group of mathematicians in the context of artificial intelligence (AI) [1–8]. The need to practically apply the task-based approach for automated problem solving in AI necessitated a more detailed examination of the very concept of a "task". This examination resulted in formulating the concept first at the methodological level and then at the formal, mathematical level. This led to the creation of an original logical-probabilistic theory of the task-based approach — **semantic modelling** [9–24, 39–45]. According to the task-based approach, a task is considered defined if and only if its formulation includes the following components:

- **Specification of the domain**, captured in the form of a model, including a description of the signature and structure of a domain-specific language (ontology), as well as general knowledge and domain-specific knowledge: initial data, facts/precedents, rules, constraints, and hypotheses [46];
- **The query (question)** that the task poses to the domain model — i.e., the specific question to which we must obtain an answer (the task's solution);
- **A criterion for query satisfaction** — specifying under what conditions we can consider that an answer (solution) to the query (question) has indeed been obtained;
- **The context** in which the answer (solution) to the query (question) should be sought and interpreted — namely: what goal we pursue by solving the task (i.e., what we expect from the obtained result and what its consequences are); what to do if the answer turns out to be negative; other conditions surrounding the task — resources, constraints, norms, etc.

Based on the specified structure of the concept of a task within the task-based approach, the following scheme of actions for need satisfaction was proposed:

BEGINNING OF THE SCHEME

STEP 1. Identification and analysis of needs.

An expert and domain analyst studies, clarifies, and conceptualizes the emerging weakly structured need, using the possibility to apply one or another existing action template that would allow satisfying it. If no suitable "template-based" method for satisfying the need is available, the associated contradiction between the desired and the actual state of affairs is identified and studied. This contradiction constitutes the true cause and content of the task to be solved.

STEP 2. Planning.

A plan of action is developed, including: creation of a model of the domain to which the task belongs; description of the query (queries) requiring an answer(s); formulation of a criterion for

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

successfully overcoming the identified contradiction; analysis of the conditions for organizing the task-solving process, treating the results obtained at this stage as the initial version of the task's context.

STEP 3. Formation of a vision (concept) for the task formulation and solution project.

In natural language, general and specific evaluative knowledge relevant to the stated task(s) is described. Relevant facts and other precedents are collected, and an ontological model of the problem domain is constructed (including concepts, relations, properties, etc.). A class of possible queries to the problem domain is formulated in general and ontological terms — these queries reveal the content of the overall task and its subtasks. A version of the task solution criterion (for both the main task and subtasks) is also formulated. The part of the task representation and domain understanding that requires either: external calculators acting as oracles, or "basic" entities already obtained or acquired manually during the project execution (which, incidentally, can also be treated as oracles) is identified.

STEP 4. Formalization of a semantic domain-specific language (sDSL) for the domain and task formulation in terms of this language.

A formal model of the problem domain is constructed, and the query and task solution criterion are formulated in the same formal terms — taking contextual conditions into account. The identified set of missing oracles is programmed, and communication with existing oracles is addressed. If a solution criterion based on testing with semantic test cases is chosen, their formal specifications are written.

Note that if STEP 4 is carried out within and using the tools of semantic modelling, we can immediately obtain a computer-based version of a system oriented towards task solving (no-code).

STEP 5. Testing and debugging.

STEP 6. Operation (deployment and use).

END OF THE SCHEME

It should be noted that the formal refinement of the concept of a task within the task-based approach required the creation of an original semantic model of computability. Its mathematical foundation is the concept of formula definability on constructive models [38], using oracle (relative) computability. As a formal language for task specification, a sufficiently expressive fragment of the predicate calculus language was chosen — the class of Δ_0-formulas and Δ_0-terms, incorporating oracle computability. This allowed restricting the scope to the class of polynomial-complexity computations without losing the expressive power of the language or its potential completeness. It has been shown that the Δ_0-fragment of a many-sorted first-order predicate calculus language corresponds to the concept of polynomial computability — in other words, polynomial computability is Δ_0-formula definable [17, 19–23]. Moreover, such a language of executable specifications can be effectively enriched with probabilistic constructions, which significantly expands the expressive capabilities of the declarative specification language [40–45].

Now, a few words about the second component of task-based engineering — **engineering**, which combines two mechanisms for interacting with generative AI models: prompt engineering and context engineering. Currently, **prompt engineering** is a discipline at the intersection of psychology, linguistics, and computer science. It studies methods for developing and optimizing queries (prompts) to ensure effective interaction with generative AI models — particularly with large language models (LLMs) — when solving various tasks. This interaction mechanism enables guiding the generative model's work toward producing accurate, relevant, and useful results through carefully crafted query formulation. Let us recall that generative models generate text by predicting the most probable next tokens (words or symbols) based on the input prompt and the data they were trained on. Therefore, the goal of prompt engineering is to control these probabilities by setting the context, constraints, and reasoning direction of the generative model. The structured nature of a prompt is especially important for its effectiveness. This means the need to specify the query processing context — for example, a role, clear instructions, examples, an explicit indication of the required answer, etc. Moreover, creating a high-quality prompt often follows an iterative process. This implies conducting a certain cycle of interactions with the generative model: formulating, test-

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

ing, analyzing the results, and optimizing. It is extremely useful to compile and use prompt libraries — collections of proven queries grouped by topics or styles.

To date, a wide range of techniques has been developed to improve the quality of generative model responses. Here are some of them:

- **Zero-shot prompting** — the model provides a response without using any examples; it is applied to solve simple tasks or tasks that are well-known to the model.
- **Few-shot prompting** — several task execution examples are included in the prompt, which helps the model better understand the requirements.
- **Chain-of-Thought (CoT)** — instructing the model to generate step-by-step reasoning before providing the final answer; this technique is especially effective for logical, mathematical, and multi-step tasks.
- **Self-Consistency** — the generative model generates multiple reasoning chains and selects the most frequently occurring answer, thereby increasing the reliability of the result.
- **Tree-of-Thoughts (ToT)** — the model explores several parallel reasoning paths simultaneously, which is useful for complex tasks.
- **ReAct (Reason and Act)** — the model alternates between reasoning steps and using external tools (information retrieval, code execution, etc.).
- **RAG (Retrieval-Augmented Generation)** — linking responses to factual data retrieved from external sources, which helps reduce the likelihood of errors.
- **Metaprompting** — the initial query generates a more detailed sub-query, allowing the model to refine the task.
- **Role-based prompting** — assigning a specific role to the generative model (e.g., expert, editor, guide, etc.), which helps tailor the style and content of the response.
- **Step-back prompting** — the generative model first considers a general question related to the task and then uses the acquired knowledge to solve the specific problem. This can mitigate bias and improve response quality.

Currently, prompt engineering as a discipline continues to actively evolve. Among the current trends is the development of **multimodal prompts**, which combine text with images, audio, and video.

As for **context engineering**, this branch of engineering is responsible for the systematic work with the context of interaction between the user and the generative model — in order to improve the quality, relevance, and security of the generated responses. Unlike prompt engineering (which focuses on query formulation), context engineering manages the ***query's environment***: dialogue history, metadata, external sources, constraints, and other elements. As in the case of prompt engineering, context engineering has also accumulated a substantial set of techniques by now. The main ones include:

- **Dialogue history management**;
- **Thematic segmentation;**
- **Role labelling (user/assistant) for clear attribution;**
- **External context retrieval (RAG);**
- **Hybrid search (keywords + semantics) to improve accuracy;**
- **Re-ranking of results before feeding them into the generative model;**
- **Context structuring;**
- **Reliability control;**
- **Context personalization;**
- **Context constraints;**
- **Dynamic context management;**
- **Context visualization;**
- **Meta-context** in the form of explanations provided to the model about the context;

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

- **Context automation** using preprocessing, postprocessing, and context agents.

The range of tools and technologies used in context engineering is also extensive, including various **vector databases, RAG frameworks, context management systems, and annotation and quality monitoring tools**. In general, the role of context engineering lies in transforming "raw" data into structured knowledge. It is important to note that the optimality of a context engineering strategy is ensured by the iterative nature of interaction with the model and the ability to adapt it to a specific task.

It should also be emphasized that prompt engineering and context engineering do not compete but complement each other, forming a unified control loop for interacting with generative models. Their relationship can be figuratively described as **"form through content**", where: prompt engineering provides the form of interaction (how to ask); context engineering provides the content — what information to feed the model, where to retrieve it from, how to structure and update the data. Thus, the prompt acts as a kind of "command line", while the context represents the data needed to obtain an accurate response. To date, several mechanisms for integrating prompts and context have been developed. The most popular ones include:

- **Explicit inclusion of context in the prompt**, where the context simply becomes part of the input query;
- **Dynamic context loading via RAG** (Retrieval-Augmented Generation), where the prompt is formulated as a special query. The system automatically retrieves the context, and it is embedded into the final prompt before it is sent to the generative model;
- **Multilevel management**;
- **Iterative optimization**, where the prompt initially sets the query, the context module loads relevant data into it, the generative model generates a response, and this response is then analyzed for quality. If needed, the prompt or context is adjusted accordingly;
- **Role separation in multi-agent systems**.

As for **agent-based AI**, research in this field began almost from the very inception of AI and was closely linked to cybernetics, automata theory, and other scientific disciplines that model the behaviour of artificial and biological entities in an external environment. In fact, the topic of agents and multi-agent systems (MAS) almost immediately split into two directions: the area related to the creation of **software agents** and communities built from them, operating in the digital world and the area of **embodied agents** and their communities functioning in the real, physical world.

Regarding the first direction, the original idea of an intelligent software agent emerged as an attempt to intellectualize the user interface. Instead of requiring the user to initiate actions via commands, it was proposed that a computer intermediary should collaborate with the user in solving tasks. Gradually, this concept expanded beyond the scope of interfaces and began to focus on: artificial intelligence methods; the use of computer networks; distributed databases; distributed computing. In the 1980s, autonomous mobile robots capable of acting without central control were also actively developed. These robots responded to environmental changes rather than to pre-defined commands. In the 1990s, the so-called **agent-oriented approach** was finally established in AI. Within this framework, attempts were made to describe a social perspective on the organization of computation — one based on the interaction of software agents during the computational process.

With the development of machine learning and deep learning at the beginning of the current century, the concept of an intelligent software agent acquired new meaning. This is because next-generation language models have learned to maintain context, reason, and interpret queries. Starting in 2023, LLMs (Large Language Models) began to be used not as a tool, but as an element of a reasoning software agent capable of independently setting goals, formulating subtasks, and executing them. This gave rise to the concept of an **AI-agent** — an autonomous system able to: understand complex goals; independently plan the steps required to achieve them; actively interact with the surrounding digital environment. It should be noted that AI-agents differ from chatbots and intelligent assistants/helpers/co-pilots in that they can not only respond to queries, but also perform actions using various tools (APIs, databases, external services). Modern AI agents are built from modules, with the main components being: a reasoning core ("brain") based on an LLM (Large Language Model), LMM (Large Multimodal Model), or VLM (Vision-Language Model); a prompt instruction; memory with context; tools for acting in the external digital world.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

AI agents are now being used to build so-called **multi-agent systems** (MAS) — communities in which several AI-agents interact with each other to solve tasks that are too complex for a single agent. The interaction can be: cooperative — for example, a team of bots collaboratively writing a report; competitive — for example, when agents negotiate the price of a delivery or compete in finding vulnerabilities. Recently, a direction has been actively developing that involves creating and using small language models (SLMs) instead of large universal models. This is associated with the development of specialized AI agents for specific tasks and domains. This has enabled the creation and development of a wide range of industry-specific solutions: AI agents specialized in finance, retail, healthcare, logistics, and so on. Various AI agent ecosystems are now being actively created, including **frameworks** and **platforms** for building and managing multi-agent systems. Generative AI models are increasingly being used as the cognitive core of AI-agents. Significant efforts are now being made to develop and implement AI safety standards, including the concept of trust, risk, and security management for AI agents and multi-agent systems. This focus on trust and security is due to the fact that generative models are prone to generating "hallucinations". AI-agents may also: perform unauthorized actions; contribute to data leaks; participate in cyberattacks; make unexpected decisions; act in ways that are difficult to anticipate in advance — especially in dynamically changing conditions. At the same time, it should be taken into account that legal responsibility still lies with a human or an organization.

Regarding multi-agent systems (MAS), they also come with their own set of challenges. For example, there are problems related to the complexity of integrating AI agents with various information systems and corporate processes. This requires complex access and control configurations. Moreover, as the number of interacting agents increases, the complexity of coordinating their actions grows exponentially. The topic of multi-agent systems has recently gained particular relevance, as it is expected that in the future, MAS will become even more widespread and will tackle increasingly complex tasks. It is anticipated that: **hyperpersonalization** will develop, based on analysis of the user's or customer's emotional state; **proactive service** will become ubiquitous — where assistance is provided before a problem arises. In the long term, the emergence of **personal AI-agents** — digital secretaries — is possible. Such agents would: manage schedules; handle correspondence; filter information flows. However, this will require addressing ethical, legal, and technical challenges related to: control; responsibility; trust; security.

Previously, the topic of software agents and multi-agent systems (MAS) composed of them was discussed. Equally relevant is the topic of **embodied and cyber-physical AI- agents**, as well as the corresponding multi-agent systems. Hereafter, by an embodied agent we will mean AI systems that interact with the physical world through their sensors and actuators. This enables them to perceive the environment using video cameras, microphones, sensors, and other devices; make decisions based on the data received; and perform actions in the real world. The architecture of an embodied agent typically includes the following modules:

- **Perception module**, responsible for image processing (CV — Computer Vision), audio analysis (ASR — Automatic Speech Recognition), and interpretation of data from other sensors (temperature, pressure, distance, power supply, etc.);
- **Understanding module**, which handles tasks such as: building an environment map (SLAM — Simultaneous Localization and Mapping); identifying objects and events; predicting changes;
- **Reasoning module**, which performs: action planning (e.g., pathfinding, optimization); decision-making under uncertainty; learning (RL — Reinforcement Learning, simulation-based learning);
- **Action module**, including: motor/actuator control; communication with other agents; adaptation to environmental changes;
- **Memory and Learning module**, responsible for: storing experience (trajectories, successful strategies); updating models through feedback.

A subclass of embodied agents are **cyber-physical agents** — embodied agents integrated into cyber-physical systems (CPS) that combine: computational components (algorithms, models); physical elements (sensors, mechanisms); network connections (IoT, 4G/5G). It is easy to see that embodied agents differ from software agents primarily in that they interact directly with the physical environment and therefore must account for the laws of physics (friction, gravity, inertia, temperature, etc.).

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

Moreover, they often need the ability to process data in real time. Finally, they are typically subject to heightened requirements for reliability and security.

Multi-agent systems (MAS) with embodied agents have several distinctive features compared to software-based MAS, due to the fact that they operate in the physical world. For example, MAS agents must: "understand" and account for their physical constraints (battery life, mechanism wear, weather conditions); be aware of each other's locations and of obstacles; synchronize timing when coordinating actions; account for potential communication limitations, instability, and data transmission delays.

Multi-agent systems (MAS), both software-based and embodied, are characterized by various types of organizational interaction architecture:

- **Centralized,** where the system designates an agent that acts as a single coordinator (the term "orchestrator" has become popular recently), assigning tasks to the other agents in the system.
- **Decentralized**, where agents negotiate directly with each other. This architecture is recommended for dynamic operating environments.
- **Hybrid**, which combines features of the two previous types of agent community organization. This is usually achieved by enabling the creation of local groups (clusters) with their own coordinators, along with the presence of a global supervisor/orchestrator.

In recent years, the **role-based organizational** architecture of MAS has become very popular. In this approach, agents are pre-assigned specific roles and competencies. Below is one possible variant of a step-by-step scheme for creating a software agent with specified competencies.

BEGINNING OF THE SCHEME

Step 1. Define the role, goals, and functions of the software agent.

Example:

Role: Financial analyst.

Goal/Task: Assess the profitability of a business plan.

Functions: Calculate NPV, IRR, payback period; perform sensitivity analysis for interest rate changes.

Step 2. Design the agent's competencies, classifying them as basic, domain-specific, or cognitive.

Step 3. Select the technological base. This involves choosing either: a universal option — typically referring to LLM agents; specialized models; hybrid systems (LLM + scripts).

Step 4. Training and configuration. The main methods for designing the software agent here are prompt engineering and context engineering.

Step 5. Integrate the AI agent into a multi-agent system (MAS). At this stage, the interaction mechanisms among the agent community members must first be defined. This is largely determined by: the class of tasks the community solves; its organizational architecture; the role composition of the agents.

When integrating a previously created AI-agent, it is usually assumed that the MAS already includes a special agent called a "**coordinator**" or "**orchestrator**", responsible for task decomposition and subtask distribution among agents. Typically, a data bus is used for task distribution.

Step 6. Testing and validation. At this stage, a specific testing scenario is selected and implemented — for both individual agents and the entire MAS. These may include, for example: **functional testing; load testing; stress testing**.

Step 7. Deployment and monitoring. This is one of the most labour-intensive and critical steps. For this reason, it is recommended to use ready-made tools that help implement and configure: orchestration; containerization; monitoring; logging.

Step 8. Iterative improvement. This step is a cyclical procedure aimed at enhancing the performance quality of both the agents and the MAS as a whole.

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

END OF THE SCHEME

Previously, we mentioned the special significance and importance of the orchestrator agent in a multi-agent system (MAS). This agent is typically assigned the following functions:

- Accepting and decomposing the initial task;
- Distributing the subtasks (obtained through decomposition) among the system's agents;
- Appointing a leader and support agents for a specific scenario or cluster in a hybrid MAS architecture;
- Managing communication channels. In particular, the orchestrator is responsible for configuring the message bus or API gateways through which agents exchange events. This may involve either: a centralized task queue; a distributed list where each agent subscribes only to the topics relevant to it;
- Tracking changes in external conditions (new data, timeouts, errors from individual agents) and adjusting priorities accordingly — determining which agent should respond first, and which responses can be collected in a "batch" later;
- Monitoring and redistributing resources to ensure the system remains responsive and reliable;
- Collective verification through cross-checking schemes: validating the main solution using a verifier agent and an arbiter agent; if discrepancies with predefined criteria are detected, activating fallback mechanisms — e.g., triggering "Plan B": redirecting the task to a backup agent or initiating retraining;
- Monitoring and reporting, which involves: collecting performance metrics for each agent (response time, accuracy, load); visualizing cluster status; automatically generating notifications about critical events, system changes, or when key indicators approach threshold values.

Thanks to orchestration, a multi-agent system can operate as a single "living" organism — where agents do not merely execute their functions sequentially, but respond synchronously and adaptively to system challenges. When designing an orchestrator, it is first necessary to clearly define: what class of tasks the MAS (and thus the orchestrator) will solve; which agents will participate in the system; what interaction scenarios with the user and between agents need to be supported; what organizational architecture of the MAS is selected. Once decisions are made regarding the task class, architecture type, and role composition of the MAS, the following are defined: decision-making algorithms; rules for task decomposition and distribution; error handling mechanisms; adaptation mechanisms for changing conditions. Here, one can use a logical-mathematical approach to build the "brain" of the orchestrator agent, or rely on a specific LLM (Large Language Model). A hybrid approach can also be applied, combining a logical model with a generative model. To date, a wide range of tools has been developed for creating the "brain" of orchestrators.

Previously, we defined task engineering as a synergistic integration of the task-based approach, engineering, and agent-based AI. In this framework, the task-based approach occupies a central place as the core of a new scientific and engineering discipline, while the other components serve as tools. Let us recall that the main problems that task engineering aims to address are:

a) Automating the transition from an unstructured need to a clear formulation of the corresponding task;

b) Automating the task-solving process itself — especially when dealing with complex, large-scale, and multi-component tasks. Solving such tasks may require: preliminary **decomposition** into subtasks; **planning actions** to address them; aggregating these partial solutions into a final solution for the entire task; accounting for necessary resources (computational, temporal, intellectual, etc.).

In addressing these fundamental problems, the **task-based approach** provides the methodological foundation for action. It specifies the direction for transitioning from an unstructured need to a structured task, and: establishes a conceptual system in which the question (requiring an answer) is formulated; fixes the task completion criterion (including quantitative and qualitative metrics); defines the logic for result verification; ensures process reproducibility. Thus, the task-based approach

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

defines the overall **goal-oriented direction** of the entire process of satisfying the original unstructured need. The other components of task engineering serve as a "bridge" or "interface" between humans and AI systems — acting as sophisticated, yet merely instrumental tools. **Prompt engineering** enables one to clearly pose and even formalize a query to generative models, specifying the structure and format of the output. It can also guide the cognitive processes of a generative model — for example, by using Chain-of-Thought or few-shot learning. **Context engineering** ensures data relevance through RAG (Retrieval-Augmented Generation). It: updates the generative model's knowledge without retraining; filters information based on reliability and freshness criteria; creates a system "memory" for sequential task solving. Its key function is to ensure precise, complete, and up-to-date interaction with generative models. **AI-agents** and **multi-agent systems** (MAS) are among the most effective modern tools for problem solving. AI-agents, with their ability to adapt to changes via a cybernetic loop (feedback), can: perform a wide range of autonomous actions (data analysis, calculations, content generation); be highly specialized in specific domains (finance, business, science, manufacturing, construction, marketing, law, etc.). As for multi-agent systems, they are ideal for solving complex and large-scale tasks because they can: scale up; support parallel and hierarchical processing of subtasks; coordinate agent interactions (orchestration); implement verification and consensus mechanisms. Overall, AI-agents and MAS can play a key role in transforming a weakly structured problem into a formalized task and in practically solving it with quality control. The direction and content of these actions are determined by the task-based approach.

Naturally, the question arises: what exactly is the synergistic effect of integrating the components of task engineering? To answer this question, let us consider one possible variant of a **complete 4-phase lifecycle for satisfying an initial need using task engineering.**

BEGINNING OF THE CYCLE

Phase 1. Problem formulation and conversion into a task (task-based approach + engineering with its prompt techniques and templates for structuring + agent-based AI). If necessary: decompose the task; build a dependency graph of subtasks; assign AI- agents to subtasks (task-based approach using a graph visualizer + context engineering with external data (RAG, knowledge bases, APIs) + multi-agent system concept).

Input: unstructured problem description.

Output: formalized task statement, decomposed into subtasks + hierarchical subtask plan for agents with dependencies.

Phase 2. Planning with smart contract generation (engineering + agent-based AI).

Input: decomposed subtask plan with dependencies assigned to MAS members.

Output: executable action plan with resources and deadlines.

Phase 3. Execution and monitoring (task-based approach + context engineering + multi-agent system concept).

Input: action plan for solving subtasks.

Output: intermediate results + progress report.

Phase 4. Adaptation and lesson capture in the knowledge base (task-based approach + prompt engineering + context engineering).

Input: monitoring data.

Output: updated plan/formulation $\rightarrow$ **Phase 3.**

END OF THE CYCLE

Naturally, each phase of this general scheme is actually a multi-step procedure that refines the sequence and content of actions for its implementation. Below is an example of a step-by-step process that reveals the content of **Phase 1.** It is assumed that by the time this step-by-step scheme is used, the basic composition of agents in the instrumental multi-system has already been formed. This composition is generally suitable for a fairly wide range of typical needs and medium-complexity tasks with clear boundaries. It usually includes:

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

- **Interviewer agent** — clarifies the initial need and requests;
- **Modeling agent** — builds a semantic model of the domain to which the need and formulated task belong;
- **Formulator agent** — generates the query for which a solution (task answer) is sought;
- **Verifier agent** — defines the task completion criterion;
- **Context aggregator** — collects data;
- **Coordinator/Orchestrator** — the main agent that manages the entire lifecycle of need satisfaction. It: initiates the activity of the appropriate agent at the right moment; if necessary, decomposes a complex task and assigns subtasks to the relevant agents; integrates solution results and other components; monitors processes;
- **Tester agent** — verifies the task before execution.

However, if we encounter a complex multi-component task, then there arises a need to expand the basic composition of the multi-agent system. For example, when launching a project to create a **new medical diagnostic service**, the following additional agents will be required:

- **Medical expert** — validates algorithms against clinical guidelines;
- **Ethical auditor** — checks for data bias;
- **Regulatory agent** — ensures compliance with standards;
- **Interface designer** — adapts the solution for doctors and patients.

If, say, the task is to optimize a **city's energy system**, the additional agents would include:

- **Power systems engineer** — models load demands;
- **Environmental agent** — assesses the carbon footprint;
- **Economist** — calculates tariffs and subsidies;
- **Crisis manager** — develops contingency plans for emergencies.

Let's return to the step-by-step process of Phase 1.

PHASE 1 START

Step 1. Initial information gathering (Interviewer agent)

Actions:

- asks the user clarifying questions using a template (context, goal, constraints, success criteria);
- records key requirements, expectations and implicit needs;
- identifies contradictions and ambiguities in the request;
- structures the information into a draft problem description.

Tools: question checklists, interview templates, active listening techniques.

Output: structured draft problem description (in Markdown, Google Docs or plain text file format).

Result: structured set of requirements (e.g., in the form of a table or numbered list).

Step 2. Semantic model construction (Modeling agent)

Actions:

- creates an ontology of the subject area (entities, relationships, rules);
- defines data types and input/output formats;
- identifies known solution patterns for similar tasks;
- builds a conceptual schema (knowledge graph, UML diagram).

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

Tools: ontology editors (Protégé), graph visualization tools (Graphviz, Neo4j Browser).

Output: semantic task model (in D0SL, OWL, RDF, UML format or as a conceptual schema).

Step 3. Formal request generation (Formulator agent)

Actions:

- transforms the draft from Step 1 and the semantic model from Step 2 into a clear system request;
- specifies the structure and format of the expected result;
- formulates an LLM prompt or API specification.

Tools: prompt engineering, technical assignment templates.

Output: formalized task (as an LLM prompt, API specification, technical assignment or machine-readable format — JSON/YAML).

Step 4. Success criteria definition (Verifier agent)

Actions:

- formulates quantitative and qualitative metrics for solution evaluation (accuracy, speed, completeness, compliance with standards);
- describes conditions under which the task is considered completed;
- creates a validation checklist and a set of test cases.

Tools: SLA templates, checklists, testing frameworks.

Output: completion criteria (checklist, set of test cases, SLA).

Step 5. Data collection and updating (Context aggregator)

Actions:

- searches for relevant external data via RAG, APIs, knowledge bases;
- filters information by reliability and recency;
- creates a "memory" for sequential task solving;
- enriches the context with data from open sources and corporate databases.

Tools: RAG systems, API clients, parsers, knowledge bases.

Output: enriched context (set of documents, dataset, source links, structured data).

Step 6. Decomposition and planning (Coordinator/Orchestrator)

Actions:

- breaks the task into subtasks considering the semantic model and success criteria;
- builds a dependency graph (which subtasks are executed sequentially, which — in parallel; "AND"/"OR" relationships);
- assigns agents to subtasks;
- allocates resources (computational, temporal);
- generates a hierarchical execution plan.

Tools: graph building tools, task planners, planning algorithms.

Output: hierarchical subtask plan with dependencies and responsible agents (in JSON, YAML format, Gantt chart or task graph).

Step 7. Validation and verification (Tester agent)

Actions:

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

- checks the plan's integrity: do all subtasks cover the original goal? Are there any contradictions?
- runs a "dry run" of the plan on synthetic data or a simplified scenario;
- identifies potential risks (data shortage, agent conflicts, bottlenecks);
- makes adjustments to the plan if necessary.

Tools: simulators, test environments, validation checklists.

Output: approved execution plan with a "ready to launch" mark and a validation report.

Step 8. Execution initiation (Coordinator/Orchestrator)

Actions:

- launches subtask execution according to the plan;
- transfers control to **Phase 2** (Planning with smart contract generation);
- records initial metrics and monitoring checkpoints.

Output: readiness signal for transition to the next phase + start-up report (status "Launch completed", start time, list of launched subtasks).

END OF PHASE 1

Key synergy mechanisms in Phase 1:

- **Sequential data transfer** — the result of each step serves as input for the next one (information is neither lost nor duplicated).
- **Agent specialization** — each agent performs a narrow task using its competencies:
  - interviewer — communication;
  - modeler — semantics;
  - verifier — metrics;
  - aggregator — data;
  - coordinator — management;
  - tester — quality control.
  - **Centralized management** — the coordinator/orchestrator synchronizes agents' work, builds the dependency graph, and controls transitions between steps.
  - **Context enrichment** — the context aggregator updates data at each step, improving the accuracy of formulations and plans.
  - **Early validation** — the tester checks the plan before execution, reducing the risk of costly errors during the execution phase.
  - **Formalization at all levels** — from a verbal request to a machine-readable plan, ensuring unambiguity and executability of the task.

Let's return to the full lifecycle of task engineering and see how the composition of a Multi-Agent System (MAS) is formed as it progresses. For example, at the stage of transition from the initial need to task formulation, the MAS included **Interviewer agent, Modeling agent, Coordinator/Orchestrator agent** and other. When moving to planning the task solution, the MAS orchestrator may additionally engage the following agents: **Financial agent** (calculates budget and return on investment); **Technical agent** (selects implementation tools); **Operational agent** (designs implementation processes); **Analytical agent** (defines success metrics). During execution of the planned solution, depending on the task being solved, the following agents may appear: **Product agent** (creates digital products); **Marketing agent** (launches advertising campaigns); **HR agent** (organizes staff recruitment and training); **Logistics agent** (optimizes supply chains). Finally, for successful task solution via task engineering, continuous monitoring of the task formulation and solution process and its adaptation to changing conditions is extremely important. For this, the following agents are needed: **Analytical agent** (collects progress data); **Coordinator** (initiates corrective cycles when deviating from KPIs); **Legal agent (**updates documents when regulations and governing documents change).

Mathematics & AI, Vol. 1, No. 2, 2026, Article 1038 DOI: 10.66693/mathai.1038
Special Issue: MathAI 2026 Selected Papers
Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

Generally, when discussing a multi-agent system involved in task engineering, it should be emphasized that its composition is not standard and must be adapted to the complexity, industry affiliation, and other specifics of the need and the task being solved. If necessary, the MAS can be expanded with specialized agents and changed during the solution process.

Summarizing the above, we first of all note the relevance and modernity of task engineering, as it reflects key trends in the development of modern AI technologies:

- **Methodological integrity** — the task-based approach provides a methodological and theoretical foundation for the new discipline, while other elements serve as practical tools;
- **Technological relevance** — the proposed integration aligns with the current trend towards specialization of AI systems (AI agents and MAS), improved interaction (prompt engineering), and knowledge management (task-based approach + context engineering);
- **Development prospects** — the proposed concept of a new scientific-engineering discipline is universal, as it can be easily scaled to new subject areas and thus may become a standard for solving complex, large-scale, interdisciplinary tasks;
- **Practical applicability** — according to the author, task engineering will be especially effective in construction, industry (smart factories), business analytics, logistics (autonomous warehouses), science, medicine (robotic surgeries), public administration, and urban management (smart cities).

The author is confident that the proposed synergistic integration of the task-based approach, prompt and context engineering, AI-agents, and MAS can form the basis of a **universal instrumental-technological task-solving system**, as it allows scalability from individual local tasks of various industry focus to global ecosystems, where: the task-based approach sets the structure and success criteria; prompt engineering ensures precise interaction with models and agents, including through cross-verification of agents; context engineering helps adapt to environmental changes in real time, thus guaranteeing knowledge relevance; MAS, consisting of software and embodied agents, are capable of implementing solutions to complex tasks in the digital and physical world.

The creation and subsequent development of the platform is conceived by the author as the development and implementation of a set of measures, on one hand, to further improve the methodological and technological foundations of the new scientific-engineering discipline — task engineering, and, on the other hand, to design and subsequently implement the instrumental-technological platform itself. Within and through this platform, in accordance with the methodology and technology of task engineering, specialized AI systems will be created to meet various classes of weakly structured needs (business, finance, medicine, logistics, construction, science, production, public administration, smart cities, etc.).

REFERENCES

REFERENCES

[1] Vityaev E. E., Goncharov S. S., Sviridenko D. I. On the task-based approach in artificial intelligence // *Siberian Philosophical Journal*. 2019. Vol. 17, No. 4. P. 5–25.

[2] Vityaev E. E., Goncharov S. S., Sviridenko D. I. On the task-based approach in artificial intelligence and cognitive sciences // *Siberian Philosophical Journal*. 2020. Vol. 18, No. 2. P. 5–29.

[3] Goncharov S. S., Vityaev E. E., Sviridenko D. I. Problem-solving approach in artificial intelligence // *Applied Mathematics and Fundamental Informatics*. 2020. Vol. 7, No. 2. P. 4–9.

[4] Sviridenko D. I., Vityaev E. E. A task-based approach to artificial intelligence and its theoretical and technological base // *Proceedings of the 18th National Conference on Artificial Intelligence with International Participation (CII-2020)* (October 10–16, 2020, Moscow, Russia) / ds. V. V. Borisova, O. P. Kuznetsova. Moscow: MFTI, 2020. 326 p. P. 36–44.

[5] Vityaev E. E., Goncharov S. S., Sviridenko D. I. Task-driven approach to artificial intelligence // *Cognitive Systems Research*. 2023. Vol. 81 (September). P. 50–56.

Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

[6] Vityaev E. E., Goncharov S. S., Gumirov V. S., Mantsivoda A. V., Nechesov A. V., Sviridenko D. I. Task approach: on the way to trusting artificial intelligence // *World Congress: Systems Theory, Algebraic Biology, Artificial Intelligence: Mathematical Foundations and Applications. Selected Works*. Moscow, 2023. P. 179–243.

[7] Goncharov S. S., Sviridenko D. I., Vityaev E. E. Task Approach to Artificial Intelligence // *Proceedings of the Workshop on Applied Mathematics and Fundamental Computer Science 2020* (Omsk, Russia, April 23–30, 2020). *CEUR Workshop Proceedings*. Vol. 2642.

[8] Sviridenko D. I. A task-based approach to building a digital twin of an organization // *World Congress "Systems Theory, Algebraic Biology, Artificial Intelligence: Mathematical Foundations and Applications"* (June 26–30, 2023, Moscow): Abstracts. Moscow, 2023. P. 770–773. DOI: 10.18699/sblai2023-14

[9] Goncharov S. S., Sviridenko D. I. Theoretical aspects of Σ-programming // *Lecture Notes in Computer Science*. 1986. Vol. 215. P. 169–179.

[10] Ershov Yu. L., Goncharov S. S., Sviridenko D. I. Semantic programming // *Information Processing: Proceedings of the IFIP 10th World Computer Congress* (Dublin). 1986. Vol. 10. P. 1113–1120.

[11] Goncharov S. S., Sviridenko D. I. Mathematical foundations of semantic programming // *Doklady AN SSSR*. 1986. Vol. 289, No. 6. P. 1324–1328.

[12] Ershov Yu. L., Goncharov S. S., Sviridenko D. I. Semantic foundations of programming // *Lecture Notes in Computer Science*. 1987. Vol. 278. P. 116–122.

[13] Goncharov S. S., Sviridenko D. I. Σ-programs and their semantics // *Logical Methods in Programming. Computational Systems*. 1987. Issue 120. P. 24–51.

[14] Goncharov S. S., Sviridenko D. I. Σ-programming // *Translations, Series II, American Mathematical Society*. 1989. Vol. 142. P. 101–121.

[15] Goncharov S. S. Conditional terms in semantic programming // *Siberian Mathematical Journal*. 2017. Vol. 58, No. 5. P. 794–800.

[16] Goncharov S. S., Sviridenko D. I. Recursive terms in semantic programming // *Sibirskii Matematicheskii Zhurnal (SMJ)*. 2018. Vol. 59, No. 6. P. 1279–1290.

[17] Goncharov S. S., Sviridenko D. I. Logical language for describing polynomial computability // *Doklady RAN*. 2018. Vol. 485, No. 1. P. 11–14.

[18] Goncharov S., Sviridenko D. Semantic modeling and hybrid models // *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)* (Novosibirsk, Russia, 2019). P. 987–990.

[19] Goncharov S., Ospichev S., Ponomaryov D., Sviridenko D. The expressiveness of looping terms in semantic programming // *Siberian Electronic Mathematical News*. 2020. P. 380–394.

[20] Goncharov S., Nechesov A. Polynomial analogue of Gandy's fixed-point theorem // *Mathematics*. 2021. Vol. 9. Art. 2102. DOI: 10.3390/math9172102

[21] Goncharov S., Nechesov A. Solution of the problem P = L // *Mathematics*. 2022. Vol. 10. Art. 113. DOI: 10.3390/math10010113

[22] Nechesov A. Functional variant of polynomial analogue of Gandy's fixed-point theorem. *arXiv*. 2024. DOI: 10.48550/arXiv.2407.02515

[23] Nechesov A. Semantic programming and polynomially computable representations // *Siberian Advances in Mathematics*. 2023. Vol. 33, No. 1. P. 66–85.

[24] Goncharov S., Nechesov A., Sviridenko D. Programming methodology in Turing-complete languages // *International Symposium on Knowledge, Ontology, and Theory (KNOTH 24)* (September 30 – October 2, 2024): Abstracts.

Received: May 20, 2026 · Accepted: June 30, 2026 · Published: July 15, 2026

[25] Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. 4th ed. Moscow, St. Petersburg: Dialektika, 2021–2022. Vols. 1–3.

[26] Boonstra L. *Prompt Engineering*. Google, 2025. URL: www.kaggle.com

[27] Technical roadmap of context engineering in LLM: mechanisms, standards, and challenges // *AI Online*. URL: itinai.ru

[28] Context in artificial intelligence: how context understanding transforms AI // *Neurobusiness AutoPilot*

[29] Will context engineers replace prompt engineers? // *Habr*

[30] Kolmogorov A. N. *Grundbegriffe der Wahrscheinlichkeitrechnung* // *Ergebnisse der Mathematik*. Berlin, 1933.

[31] Church A. An unsolvable problem of elementary number theory // *American Journal of Mathematics*. 1936. Vol. 58. P. 345–363.

[32] Altshuller G. S. *Finding an Idea: Introduction to TRIZ — Theory of Inventive Problem Solving*. 3rd ed. Moscow: Alpina Publisher, 2010. 392 p.

[33] Altshuller G. S., Shapiro R. B. On the psychology of inventive creativity // *Voprosy Psikhologii*. 1956. No. 6. P. 37–49.

[34] Sobolev P. A. *How to Learn to Invent*. Uzhgorod: Karpati, 1973. 128 p.

[35] Polovinkin A. I. Interdisciplinary fund of heuristic techniques for object transformation // Fundamentals of engineering creativity: textbook for university students. Moscow: Mashinostroenie, 1988. 368 p.

[36] Ershov Yu. L., Samokhvalov K. F. On a new approach to the philosophy of mathematics // Structural analysis of symbolic sequences. Novosibirsk, 1984. Issue 101. P. 141–148.

[37] Ershov Yu. L., Samokhvalov K. F. Modern philosophy of mathematics: ailments and treatment. Novosibirsk: Parallel, 2007.

[38] Ershov Yu. L. Definability and computability. Novosibirsk: Bukinist, 2006. 288 p.

[39] Evgenii Vityaev. The logic of prediction. In: Proceedings of the 9th Asian Logic Conference (August 16-19, Novosibirsk, Russia), World Scientific Publishers, 2006 pp.263-276.

[40] Evgenii Vityaev, Stanislav Smerdov. On the Problem of Prediction // K.E. Wolff et al. (Eds.): KONT/KPP 2007, LNAI 6581, Springer, Heidelberg, 2011, pp. 280-296.

[41] A.V. Demin, E.E. Vityaev. Learning in a virtual model of the C. elegans nematode for locomotion and chemotaxis // Biologically Inspired Cognitive Architectures. (2014) v.7, pp.9–14

[42] Demin A.V., Vityaev E.E. Adaptive control of multiped robot // Procedia Computer Science 145C (2018) pp. 629-634.

[43] Alexander V. Demin and Evgenii E. Vityaev. Adaptive Control of Modular Robots // A.V. Samsonovich and V.V. Klimov (eds.), Biologically Inspired Cognitive Architectures (BICA) for Young Scientists, Advances in Intelligent Systems and Computing 636, Springer, 2018, pp. 204-212.

[44] E.E. Vityaev, L.I. Perlovsky, B.Y. Kovalerchuk, S.O. Speransky Probabilistic dynamic logic of cognition. Biologically Inspired Cognitive Architectures. Special issue: Papers from the Fourth Annual Meeting of the BICA Society (BICA 2013), v.6, October, Elsevier, 2013, pp.159-168.

[45] Vityaev, E., Odintsov, S. How to predict consistently? Trends in Mathematics and Computational Intelligence In: Studies in Computational Intelligence, 796, Maria Eugenia Cornejo (ed), 2019, 35-41.

[46] V.Gumirov, P.Matyukov, D.Palchunov, Semantic Domain Specific Languages, IEEE, (http://bit.ly/sDSL-RU)