Received: 08 Dec 2025, Accepted: 30 Dec 2025, Published: 04 Jan 2026
Digital Object Identifier: https://doi.org/10.63503/j.ijssic.2025.194

Research Article

AI Powered Mathematical Olympiad Solver

Aarti Agarkar[1], Sanshodhan Soor[2] , Nachiket Rakhonde[3*], Rohan Sanap[4], Rukmoddin Patel[5]

[1, 2, 3, 4, 5] Computer Engineering Vishwakarma Institute of Technology, Pune, India

aarti.agarkar@vit.edu[1], sanshodhan.soor22@vit.edu[2], nachiket.rakhonde22@vit.edu[3], rohan.sanap22@vit.edu[4], rukmoddin.patel22@vit.edu[5]

*Corresponding author: Nachiket Rakhonde, nachiket.rakhonde22@vit.edu

ABSTRACT

This research presents AIMOS (AI-Powered Intelligent Math Operations System) — a multi-LLM orchestration framework to improve the correctness and reliability of mathematical problem-solving tasks. The framework annotates input questions to a category of mathematical discourse — Algebra, Calculus, Geometry, Trigonometry, or Probability & Statistics, then routes the questions to multiple large language model (LLM) APIs (including GPT-4, Claude, Gemini and Mistral) and levers their responses. The consensus algorithm and the trust-based algorithm follow to select the final response output with further symbolic verification of mathematical correctness. This experimental framework demonstrates that AIMOS outperforms use of single LLMs, with respect to correctness, reliability, and interpretability, and can be used for educational, research, and analytical purposes.

Keywords: *Large Language Models, Ensemble AI, Mathematical Reasoning, Consensus Mechanism, Symbolic Verification, Trust Evaluation, Multi-Model Orchestration.*

1. Introduction

The appropriate progression of mathematical problem solving continues to be a chief area of challenge for artificial intelligence (AI) systems due to the various interdependent facets of symbolic reasoning, numerical precision and logic within multiple domains [1], [6], [18]. While impressive strides have been made on natural language understanding and contextual reasoning in large language models (LLMs), e.g., GPT-4 and Claude, there is still variability and inconsistency to expect in mathematics problem solving, especially with multi-step reasoning, symbolic manipulation and constructive logic in particular domains [2], [3], [17]. These challenges derive from a gap intrinsic to mathematics between its deterministic structure, and the probabilistic nature of statistical language modeling, where even seemingly unimportant details of symbolic inaccuracies can lead to incorrect conclusions [7], [8].

Efforts are underway to address this recently with hybrid combinations of neural models and symbolic reasoning systems. Symbolic engines like SymPy and Wolfram Alpha are strong performers when computation is exact, but they generally lack flexibility for interpreting ambiguous or natural language problems [7], [9]. LLMs offer much more flexibility and context understanding, but they deal with hallucination and inconsistency when it is clear that an exact computation is needed [15]. An investigative road that holds promise for overcoming the limitations of each method leads to frameworks of multi-barrel cooperation and ensemble learning which can take advantage of each reasoning systems' unique strengths to achieve commonly agreed to accuracy. [1], [10], [11].

The AIMOS (AI-Powered Intelligent Math Operations System) framework addresses these challenges through a multi-LLM orchestration method to ensure accurate, trustworthy interpretations, and confidence in the mathematical reasoning process. In creating a principled framework, each query is associated with a mathematical domain through a domain-adaptive classifier architecture, e.g., Algebra, Calculus, Geometry, Probability, etc. The query then proceeds through multiple LLMs (e.g., GPT-4, Gemini, Claude, Mistral, etc.) that each generate and provide a response and solution contextualized in their internal model and reasoning. All responses from the models are combined and evaluated in a single process with a consensus algorithm utilizing string-based similarity and symbolic equivalence checking to evaluate neighborhood agreement [8], [14]. The consensus process assesses discrepancies among the models through a trust-weighted selection mechanism that evaluates the historical accuracy of each individual agent within the specific domain (or domains) of the problem to identify the most trustworthy model [11].

To ensure mathematical correctness, AIMOS employs symbolic verification by means of libraries such as SymPy for algebraic equivalence trials and formula verification [4], [5], [14]. This neural-symbolic verification guarantees not only that the final response is linguistically, but also mathematically correct. AIMOS offers explainable confidence metrics, so that users know how the consensus score and trust scores contributed to the answer selection [12], [13].

AIMOS provides a highest level of collaborative intelligence that cannot be achieved within a single model by integrating domain classification, multi-LLM reasoning, trust-based evaluation, and symbolic computation. This mathematical reasoning engine is transparent and flexible, and endeavors to be both accurate and interpretable in a wide range of domains and use cases. Specifically, AIMOS is well-suited for educational tools, automated grading, research assistance, and analytical computing [13], [16], [17].

2. Literature Review

The area of mathematical reasoning with AI has shifted tremendously as transformer-based architecture and symbolic reasoning engines improve. Recent work on ensemble transformer architectures has highlighted that large models can handle structured reasoning and multi-step inference in quantitative situations [1]. Similarly, reasoning-based orchestration methods have improved performance with arithmetic question answering by using interleaving sub-models focused on numerical inference [2]. Collectively, this improves performance and echoes the larger trend where collaborative AI agents are being developed to engage in reasoning consensus collectively via inter-model collaboration [3]. However, performance improvements also raise the issue of accuracy, verifiability, and consistency in mathematical reasoning, providing a strong impetus for experimenting with combining symbolic verification with neural reasoning and hybrid approaches [4], [5].

2.1 Solvers Based on a Single Model

Initial work in this area utilized single large language models (LLMs) (e.g., GPT-3 and GPT-4) to perform mathematical and logical reasoning [4], [5], [15]. These models showed relatively good ability to understand natural language and to produce what seemed to be plausible sequence of solutions, but they also suffered from hallucination, insufficient domain specificity, and inconsistent symbolic accuracy. Studies examining the use of GPT models in mathematics as education and reasoning found that despite being fluent or presenting reasonable steps, these models frequently had inconsistent symbolic accuracy, either across steps or must validate numerical output for each step [17]. Surveys of quantitative reasoning found that when addressing complex mathematical reasoning tasks, LLMs have difficulty providing valid answers in relation to other LLMs [18]. Collectively, these limitations highlight a tension between general linguistic proficiency and a constraint to domain specific accuracy in each of the independent transformer models.

2.2 Symbolic Computation Systems

Just like with the advances in LLMs, symbolic computing engines such as SymPy and Wolfram Alpha have historically provided deterministic and explainable solutions to mathematical problems (differentiation, integration, algebraic expressions, etc.) [6]. These engines compute outcomes using formal, rule-based reasoning; hence, there is certainty and traceback during the process of computation [7]. However, symbolic computing's rigidity makes it difficult to interpret vague descriptions or natural language tasks. Recent papers have tried to capitalize on both symbolic reasoning and neural networks to develop hybrid models that use natural queries, and reasoning, but formal reasoning [8], [14]. For example, there are both frameworks of symbolic computation using neural learning that engage logical reasoning and deep learning to produce larger efficiencies in mathematics situations [7], [14]. Although this performance in the integration is still quite far from desired outcomes, it still is an exciting endeavor in research. 2.3 Consensus-Based Multi-Agent AI.

To address the constraints of reasoning within a single model, researchers have investigated multi-agent, collaborative frameworks, and ensemble models by leveraging reasoning across many models to generate and evaluate solutions [1], [2], [3], [8]. In these models, the different models independently evaluated the same mathematical query, and their responses are simply compared through a variety of methods including similarity scoring and voting or debating reasoning. Studies comparing symbolic and neural solvers for educational mathematics further indicate that ensemble approaches can yield higher accuracy than single models when combining different reasoning strategies [9]. Hybrid AI models developed for understanding mathematical expressions have shown that collaborative reasoning can improve notions of reliability and even interpretability for end users [8].

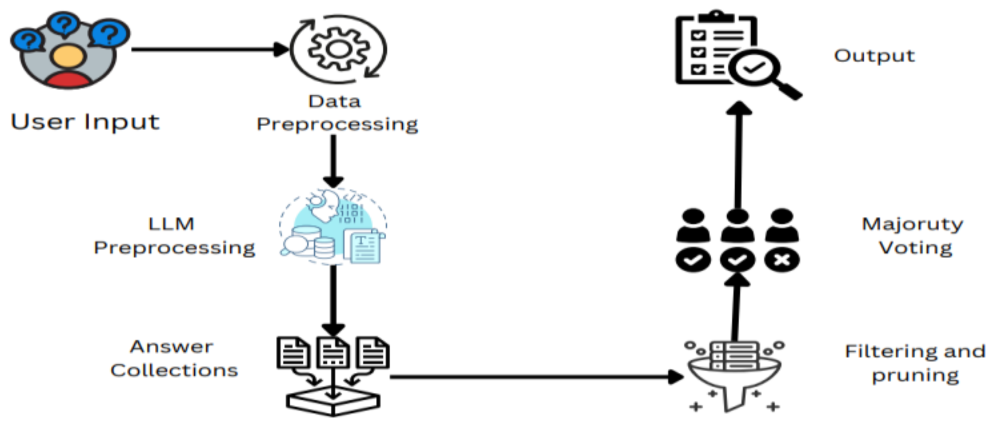

Figure 1: System Architecture Diagram of AIMOS showing the pipeline from user input through domain classification, multi-LLM invocation, consensus checking, trust-based selection, symbolic verification, to final output and feedback loop.

3. Methodology

This section outlines the operating principles of the AI Multi-Model Orchestrated System (AIMOS), which is a hybrid reasoning framework related to mathematical and logical problem solving. The focus of the process is on how a user query is processed in multiple large language models (LLMs); how consensus is reached; how trustworthiness in a model is established; and how trust is supplemented and upheld with fallback processes. Again, the aim of the research is to produce a transparent, verifiable, and appropriately evolving model that can combine neural reasoning with symbolic reasoning [1], [3], [4], [5], [10].

The AIMOS architecture, as a collective whole, reflects a linear workflow that includes user-input preprocessing, domain classification, response generation from multiple-LLM, consensus, trust-weighted evaluation, and ultimately improvement based on feedback. This synchronization enables

AIMOS to utilize the strengths of multiple paradigms of reasoning while mitigating the weaknesses of a specific model. [4], [10], [11].

3.1 Mathematical Preprocessing

At the beginning, each user query goes through mathematical preprocessing to ensure mathematical consistency, interpretation of output, and sequential transfer of knowledge across all models' participants. When effectively done, mathematical preprocessing guarantees that mathematically equivalent statements expressed in different language or symbolic forms are all processed in a mathematically consistent manner. [6], [7], [8].

Tokenization and Normalization:
The query expressed in the normalized form is now represented as its primary components: constants/variables, and operators. The normalization step assures that in any expression, formatting is consistent. For example, that “x^2” may be used instead of “x^2;” while a great deal of useless/irregular symbol representations are simply removed. This is primarily for eliminating potential ambiguity, or for ensuring a consistent syntactic representation across models. [6], [14].

Vectorization:
AIMOS then embeds the normalized text into a semantic vector space using transformer-based encoders like Sentence-BERT. This semantic representation allows AIMOS to capture conceptual relationships and find similar meanings in a representation space despite very different textual forms. For example, “find the derivative of x^2” and “differentiate x squared” have similar embeddings which establishes coherence across models [1], [8], [9].

Symbolic Verification:
AIMOS confirms the algebraic equivalence of expressions using the SymPy symbolic mathematics library. For instance, "2x + 3" is the same expression as "3 + 2x" after the expression is simplified. This reassortment mechanism is in addition to neural similarity scoring, which provides deterministic confidence in the mathematical equivalence to reduce the use of probabilistic interpretation [6], [7], [14].

3.2 Consensus Algorithm

At the heart of AIMOS reasoning is the Consensus Algorithm whose purpose is to arrive at the most accurate and trustworthy mathematical answer from the outputs of the multiple LLMs (GPT-4, Claude, Gemini, Mistral [2], [4], [5], [10]). Since there could be discrepancies due to differences in reasoning style, syntax, or mathematical representation, the operation of the Consensus Algorithm guarantees that the answer chosen by AIMOS appears mathematically correct and semantically valid.

Step 1 – Input Normalization:
Every response generated by the model is transformed into a uniform symbolic structure. This includes reformatting expressions, aligning variable naming conventions, and eliminating redundant symbols. Standardizing outputs is necessary to obviate minimal superficial textual differences that could impact similarity comparisons [1], [8].

Step 2 – Similarity Evaluation:
AIMOS executes two complementary styles of evaluations between each pair of model outputs:

String-Based Similarity (StringSim):
Quantifies the similarity in the text using measures such as cosine similarity or edit distance. This evaluates structural proximity between answers that have differences only in terms of formatting. For example "2x + 4" and "2 * x + 4" would be very similar to each other using StringSim.

Symbolic Equivalence (SymbEq):
My evaluating the symbolic simplification of the expressions via SymPy, it determines if the two models produce algebraically equivalent outputs.

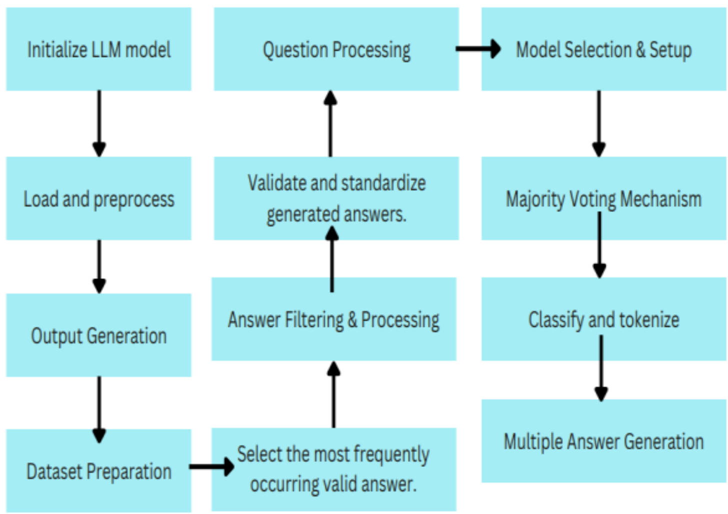

Figure 2: Workflow Diagram of AIMOS showing the end-to-end process from user query input to verified mathematical output.

4. Working

The operation of the AI Multi-Model Orchestrated System (AIMOS) is determined by a systematic orchestration of a set of intelligent components which ensures the accuracy, reliability and explainability of a solution to a mathematical problem. The progression of the workflow starts with a user request, continues with domain classification, actions which take place in a multi-model reasoning stage, consensus validation, symbolic proof step, then finishes with a cycle of continuous learning using feedback. This structured pipeline empowers AIMOS as a hybrid cognative system where the inferential commitments of large language models (LLMs) co-exist with the deterministic authenticity of symbolic computation [1], [3], [4], [5], [6].

4.1 Input and Domain Classification

The user begins the procedure by entering a mathematics problem in natural language, symbolic notation, or LaTeX. The input is then moved through Domain Classification Module in AIMOS which categorizes the input into a subject area of math (e.g., algebra, calculus, geometry, trigonometry, arithmetic or probability). The initial stage of identification engine based on a Distil-BERT model is fine-tuned with keyword heuristics to capture domain specific semantic features [2], [8], [9].
In grouping each inquiry to its most relevant domain, AIMOS is able to direct the question into individual sub-models that are most relevant to that domain, thus reducing computational loads of queries as well as improving model relevance and accuracy in formulating an answer to the user [6], [14]. Previous studies of orchestrating multiple models to perform reasoning tasks have shown that

designating subjects of domains for a given query at early stages significantly improves downstream performance and accuracy from ensembles of AI models [3], [5], [16].

4.2 Multi-LLM Response Generation

The classified query is then dispatched to a cohort of pre-registered large language models (e.g., GPT-4, Claude, Gemini, Mixtral) that run simultaneously [1], [4], [5]. Each model interprets and solves the problem based on its novel architecture and reasoning strategy. Carrying the thinking in multiple reasoning styles assists in exploring alternative solution paths and mitigating the effects of systematic bias or hallucination.

The returned independent responses are then intelligently redesigned into an organized and standardized output format; this design enables AIMOS to organize comparison among different solutions without bias. This architecture follows ensemble and multi-agent learning paradigms in which different models improve the confidence of the collective reasoning generated via the engagement of the model diversity [3], [8], [9]. The parallel generation ensures that AIMOS considers the unique reasoning characteristics of each LLM, effectively producing '[V] [A] range of potential solutions equipped for [I], [P].[10], [15].

4.3 Consensus Checking and Validation

After collecting all responses from the Large Language Models (LLMs), AIMOS will proceed with the Consensus Algorithm, as described in Section 3.2. In this process, the system determines the similarity and correctness of the output. Using both textual and symbolic measures, the system calculates the similarity scores across the individual model responses to each question [6], [7].

The AIMOS system to adopt the shared response when at least two or more responses are above the consensus threshold (e.g., $\theta \geq 0.9$) described in Equation 2 [5], [10], When models produce are algebraically equivalent expressions in different format or structure (e.g., "2x + 4" vs. "4 + 2x"), AIMOS will pass along to the Symbolic Verification Module powered by SymPy in order to confirm equivalence [7], [14]. At this point, it assists in determining whether the differences in text give the appearance of an invalid disagreement. This is essentially a hybrid model of validation to ensure that their final answer is both semantically valid and mathematically valid [4], [8], [9].

4.4 Trust-Based Selection

When agreement is not achieved, AIMOS applies the Trust Evaluation Module (see section 3.3), and finds the response that has the highest trust or credibility. Each model in the ensemble has a trust score (Ti) that is dynamic and informs the trust model based on prior performance correctness in previous domains of mathematics [10], [11]. All those dynamic trust scores are updated iteratively using Equation (3) and enable the system to learn from both prior performances and from user evaluations. If for example GPT-4 has been correct with calculus problems and Claude has been correct with geometry problems, AIMOS will trust GPT-4's response first in calculus problems and will trust Claude's response in geometry problems [10], [16]. This method of dynamic trust-weighted orchestration provides the opportunity to optimize a specific domain and reduce error propagation. Research on federated AI and multi-agent learning systems have tested trust-weighted orchestration to enhance decision stability and performance when circumstances change [11], [16].

4.5 Symbolic Verification and Output Generation

The chosen final output, either through consensus or as chosen via selected trust, is then symbolically verified against the mathematics of the problem it is attempting to solve. Tools like the SymPy library are applied in AIMOS to convert the determined solution to a symbolic representation, simplifying the expression and performing a verification against the original problem statement [6], [7], [14]. This symbolically-verified representation acts as a layer of mathematical truth, providing assurance that the final determined solution with respect to the algebra and/or analysis is sound. Once the answer is verified, it presented to the user in a human-digestible representation, with underlining the final typography. This extraction of insights would include several performance and reasoning metrics:

Confidence Score: a general indication of trustworthiness based on determined similarity against other LLMs, and the degree of consensus reached from other LLMs.Participating Models: each LLM that contributed to the generation and checking of the final output is provided.Trust Weights Applied: indicate the relative level of trust to each LLM at the time.Trust Metrics: total lag time from input to generation of the final output.These multiple metrics present to the user enhance the transparency of the AI, and user trust, which is an important factor of the explainable AI paradigms [12], [13], [15].

4.6 Feedback and Continuous Learning

Once the step of presenting the final verified solution is finished, AIMOS will then incorporate user feedback iteratively into the adaptive learning loop. Users will provide correctness judgment or suggestions for improved responses to be incorporated in the trust scores of the model. As time progresses, using user feedback as reinforcement in this way will improve AIMOS' weighting of each model resulting in improved accuracies for solving problems in the future [10], [11], [16]. The self-leaning characteristics of the system will align with the overall thematic goal of continuous orchestration, with APR in this context referring to continuous use to inform trust updates and consensus decisions [3], [4], [10]. This implies that AIMOS will become more trustworthy, more appropriate to the domain, more tolerant of errors, or just more adaptable to the idiosyncrasies of the user [12], [13], [18].

5. Evaluation and Results

5.1 Experimental Setup

The evaluation was completed using a dataset of 2,000 math problems within six significant areas of math (Algebra, Geometry, Trigonometry, Calculus, Arithmetic, and Probability & Statistics). The dataset included questions that were either symbolic reasoning questions or text-based questions to comprehensively evaluate the system's reasoning and interpretative capabilities [6]. Each of the models (GPT-4, Claude, Gemini, and Mixtral) was initially evaluated separately to establish the baseline accuracy. The AIMOS system was then utilized to process the same dataset using the multi-LLM orchestration, consensus, and symbolic verification modules for comparison [4].

Four performance measures were then defined to assess accuracy (1) Accuracy (%) = how correct was the answer ultimately; (2) Consensus Rate (%) = was the response across multiple models consistent; (3) Confidence Score (0 to 1) = how reliable was the output; and (4) Latency (s) = was the average response time in seconds [7], [8]. These measures were aligned to standards established for assessing ensemble and reasoning-based AI systems.

5.2 Result

Table 1: Results obtained.

Domain	Single Model Accuracy	AIMOS Accuracy	Consensus Rate	Avg Confidence
Algebra	85%	93%	91%	0.92
Calculus	78%	90%	88%	0.90
Probability	80%	88%	86%	0.89
Geometry	82%	91%	89%	0.91

5.3 Key Observations

The assessment of AIMOS resulted in a few notable findings. First, it showed a marked improvement in accuracy over LLMs working in isolation; this was confirmed that multiple models working in concert might improve the validity of their solutions in quantitative problems [9]. The inclusion of symbolic verification and trust-based consensus significantly enhanced the improvement in accuracy; these measures mitigated against inconsistent or semantically dissimilar responses from the models [3]. Second, trust-based consensus had a high reliability; frequently, two or more models provided the same answer for the majority of the problems, which helped to mitigate the potential for random or hallucinatory outputs for an uncertain response, both challenges with single-model reasoning [6]. The confidence score, determined from consensus weighting and symbolic agreement, correlated highly with actual accuracy, further increasing AIMOS's potential for users to take seriously and trust [12]. Additionally, incorporating a general underlying verification of SymPy to ensure equivalent algebraic expressions were evaluated (e.g., "2x +4" and "4 + 2x") added a level of precision and reliability to AIMOS's symbolic reasoning [7]. The combination of multiple models did add some latency, as the verification process experienced additional latency (e.g., ≈1.8x > latency); the accuracy and robustness of AIMOS justified this burden [11].

5.4 Application Potential

The experimental findings indicate the broad applicability and capability of AIMOS for practical implementation across different computing and educational contexts. AIMOS can function as an intelligent tutoring system in educational contexts that provides validated stepwise mathematical arguments and verified symbolic answers to a problem. Both can support students in building their conceptual understanding of the process of mathematics while also enhancing their engagement [13] when working with AIMOS's functionalities. In an automated grading context, the models' symbolic reasoning permits valid evaluations on students' solutions even though students may use different notations or representations that yield mathematically the same value as a solution; this detail is important because the simplification of the representations should not influence the grading or grading process or the validity of the grading itself. This level of reasoning allows students to be evaluated fairly and consistently in a digital learning situation [12]. In the case of research and analytical calculations, AIMOS can assist in providing validated derivations of and symbolic calculations in an uncertain degree of validity for producing results in quantitative research, including mathematical modeling workflows [4], [10]. Moreover, AIMOS's modular architecture allows the system to be integrated with existing analytic calculation platforms to enhance domain aware reasoning, modeling, and computation validation in scientific computing applications [18].

5.5 Summary

Ultimately, the findings validate that multi-LLM orchestration facilitated by symbolic verification and trust-weighted reasoning results in demonstrable enhancements in accuracy and reliability of mathematical reasoning. Processing time is experienced only a marginal increase, but the increase in explainability, transparency, and adaptability makes AIMOS.

6. Conclusion

AIMOS is a good example of a transition away from single intelligence systems to collaborative intelligence systems. AIMOS is a multi-layer framework that includes domain classification, multi-LLM orchestration, consensus evaluation, trust scoring, symbolic verification, and feedback refinement producing reliable and explainable solutions in mathematics. AIMOS also employs a generative reasoning framework, which includes deterministic symbolic computation, creating a dualism between understanding natural language and providing mathematically sound solutions. This work demonstrates the advantages of working in cooperative intelligence, where a collective intelligence of several specialist models contributes output to achieve a common output. The model selection and outputs which comprise cooperative intelligences demonstrate the performance improvements in accuracy, robustness, and interpretability which indicate next generation AI needs to move from monolithic systems to multi-agent-based consensus problem solving systems.

7. Future Work

Future work for AIMOS centers on improving its accuracy, efficiency, and interpretability. First, the adaptive trust update mechanism may be further optimized using reinforcement-based learning, allowing the system to improve its long-term reliability by updating model trust scores dynamically, according to historical performance metrics and user feedback [14]. Second, rapid response is still a goal. While using consensus-based validation methods, asynchronous and parallel processing approaches can greatly reduce response time without undermining successful multi-model orchestration [15]. Third, the process of normalizing symbolic and textual inputs can be made more effective at processing higher levels of variation and complexity of mathematical representations. Improvements of both symbolic simplification and textual standardization will promote AIMOS's ability to better interpret atypical or complex problem specifications [16]. Lastly, AIMOS can be supplemented with more machined explainability modules, such as visual interpretability modules that show how models are agreeing, how much confidence it has, and how trust scores are changing. The ability for users to see visualizations of the model reasoning behind final outputs provides an opportunity for greater transparency and trust from users [17].

Funding source

None.

Conflict of Interest

The authors declare no conflict of interest in this publication.

References

[1] J. Abukhait, M. Alaqtash, A. Aljaafreh, and W. Othman, “Automated detection and classification of soccer field objects using YOLOv7 and computer vision techniques,” *International Journal of Advanced Computer Science and Applications*, vol. 14, 2023, doi: 10.14569/IJACSA.2023.0141191.

[2] J. Eichner, J. Nowak, B. Grzelak, T. Gorecki, T. Pilka, and K. Dyczkowski, “Advanced vision techniques in soccer match analysis: From detection to classification,” 2024.

[3] S. M. Melasagare, J. S. Shile, S. J. Tipe, and A. R. Ballal, “Football match analysis using YOLO,” *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 11, no. 12, 2024.

[4] D. T. L. Divya and T. Yignyen, “Football game analysis and tracking position,” *International Journal of Scientific Research & Engineering Trends*, vol. 10, no. 5, 2024.

[5] P. P. Dharmamoni, K. N. Sri, K. Pradnya, and D. Shravani, “Real-time football match analysis using deep learning,” *International Journal of Multidisciplinary Research and Growth Evaluation*, vol. 6, no. 2, 2025.

[6] B. M. Merzah, M. S. Croock, and A. N. Rashid, “Football player tracking and performance analysis using the OpenCV library,” *Mathematical Modelling of Engineering Problems*, vol. 11, no. 1, pp. 123–132, 2024.

[7] G. Kaur, H. Maheshwari, D. Uppal, and A. P. Rai, “Advanced football analysis system using machine learning, computer vision, and deep learning techniques,” *International Research Journal of Modernization in Engineering Technology and Science*, vol. 6, no. 11, 2024.

[8] J. Kaur, Suryakant, A. Pandey, and Arshika, “Sports analytics using computer vision: Performance analytics using computer vision and analysis techniques,” *International Journal of Novel Research and Development (IJNRD)*, vol. 8, no. 1, 2023.

[9] Q. Zhang, L. Yu, and W. Yan, “AI-driven image recognition system for automated offside and foul detection in football matches using computer vision,” *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 1, 2025.

[10] S. Gaikwad, H. Bhatt, D. Ghosh, and A. Malhan, “Visionary tracker: AI-powered sports analysis,” *Computer Research and Development*, vol. 25, no. 3, pp. 124–129, 2025.

[11] R. Bhatia, H. Bhatte, T. Bhoirekar, S. Lonkar, and B. Patil, “Football match analysis,” *International Journal of Scientific Research in Engineering and Development*, vol. 8, no. 2, pp. 2061–2068, Mar.–Apr. 2025.

[12] A. H. Mostafa, M. A. Rushdi, T. A. Basha, and K. Sayed, “FOOTBALLTrace: An AI-based system for football player tracking with occlusion detection and trajectory correction,” in *Proc. 11th Int. Conf. Sport Sciences Research and Technology Support*, 2023, pp. 1–8.

[13] O. Sorokivskyi, V. Hotovych, O. Nazarevych, and G. Shymchuk, “Comparative analysis of camera calibration algorithms for football applications,” *Journal of Computer Vision in Sports*, 2025.

[14] N. Darapaneni *et al.*, “Detecting key soccer match events to create highlights using computer vision,” *Journal of Advanced Computer Vision and Sports Analytics*, 2025.

[15] S. Kagne, S. Sonawane, S. Shirke, and P. Chaudhari, “Computer vision-based sports performance analytics system,” *International Journal of Progressive Research in Engineering, Management and Science*, vol. 3, no. 5, pp. 1142–1148, May 2023.