National Seminar on Applications of Mathematics in Modern Science and Technology (20th May-2026)
Organized by: S.R. Government Arts & Science College
Kothagudem, Telangana

International Journal of Science Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

Advancing Mathematics Research with AI-Driven Formal Proof Search

B. Elizabeth Rani, Assistant Professor
Dept of Mathematics, S.R.R.Govt Arts and Science College (A), Karimnagar

Abstract- Large language models have shown remarkable promise in solving complex mathematical problems, yet their tendency to produce plausible but logically flawed reasoning—known as hallucinations—has long limited their utility in serious research. This paper reviews a landmark 2026 study by George Tsoukalas and nineteen other researchers from Google DeepMind and affiliated institutions, which demonstrates how combining large language models with formal proof verification can overcome this limitation. The study introduces a framework called AlphaProof Nexus that autonomously resolved nine open Erdős problems, proved forty-four conjectures from the Online Encyclopedia of Integer Sequences, and contributed to ongoing research across combinatorics, graph theory, algebraic geometry, and quantum optics. This paper provides a conceptual, equation-free explanation of the study's methodology, its key findings, and its implications for the future of AI-assisted mathematical discovery.

Keywords- AI-Driven Formal Proof Search, Automated Theorem Proving (ATP), Formal Mathematics, Mathematical Reasoning, Machine-Assisted Proofs

I. INTRODUCTION

In recent years, large language models have achieved remarkable success in mathematics, solving competition-level problems and even assisting with research-level theorems. However, a fundamental obstacle has prevented their widespread adoption in serious mathematical research: unreliability. The same language model that can produce an elegant proof of a geometry problem can just as easily generate a superficially convincing but logically flawed argument—a phenomenon researchers call hallucination. Because these errors can be subtle and difficult to detect, any natural-language proof produced by a large language model requires expensive expert review before it can be trusted. Mistakes that go unnoticed in intermediate steps can cascade through a proof, severely limiting the complexity of tasks that can be safely delegated to artificial intelligence.

A promising solution to this problem is to use large language models to generate proofs in formal languages—rigorous, computer-readable languages such as Lean, in which a compiler automatically verifies every logical step. If a proof is written in such a language, there is no ambiguity: either the compiler accepts it as correct, or it rejects it. This approach eliminates the risk of hidden logical errors, because the verification is performed automatically and exhaustively by a machine.

This paper reviews a 2026 study by George Tsoukalas, Anton Kovsharov, Sergey Shirobokov, and seventeen other co-authors, most affiliated with Google DeepMind, that represents the first large-scale evaluation of this method's ability to solve open, research-level problems. Their findings demonstrate that artificial intelligence, when properly integrated with formal verification, can autonomously make

International Journal of Science Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

genuine contributions to mathematical research—resolving problems that had remained unsolved for decades, at a cost of only a few hundred dollars per problem.

II. THE CHALLENGE OF AI UNRELIABILITY IN MATHEMATICS

To appreciate the significance of this work, one must first understand why large language models are inherently unreliable for mathematical reasoning. Language models are trained to predict the most likely next word in a sequence, based on patterns in their training data. They have no internal mechanism for checking logical consistency. When asked to prove a theorem, they produce text that looks like a plausible proof, but they cannot guarantee that each step follows logically from the previous ones. In the terminology of artificial intelligence research, language models lack a built-in "grounding" in formal logic.

This unreliability is not merely a nuisance—it is a fundamental barrier to using large language models for research. A mathematician who suspects that a proof might contain an error must scrutinize every line, effectively replicating the work of the model. For long proofs, this scrutiny can be more time-consuming than simply proving the theorem from scratch.

III. THE SOLUTION: FORMAL PROOF VERIFICATION IN LEAN

The approach taken by Tsoukalas and colleagues addresses this problem by changing the language in which the proof is written. Instead of asking the large language model to produce a natural-language proof, they ask it to produce a proof in Lean, a formal proof assistant developed at Microsoft Research. Lean is a programming language designed specifically for writing mathematical proofs. Every statement in Lean is expressed in a precise, unambiguous syntax, and every step must be justified by a logical rule. When a Lean proof is compiled, the Lean kernel—a small, trusted piece of software—checks each step exhaustively. If the proof is correct, the kernel accepts it. If there is any error, the kernel rejects it with an error message pointing to the problematic line.

This automatic verification eliminates the problem of hallucinations. The large language model can still propose incorrect proofs, but those proposals are immediately rejected by the Lean compiler. Only proofs that pass this rigorous, automated check are retained. The researcher, therefore, never sees an incorrect candidate—only verified results.

This paradigm—using a large language model to explore the space of possible proofs and a formal verifier to filter out incorrect candidates—is the central innovation of the study.

IV. METHODOLOGY: THE ALPHAPROOF NEXUS FRAMEWORK

The researchers developed a framework called AlphaProof Nexus to coordinate multiple artificial intelligence agents in the search for formal proofs. The framework supports two distinct agent designs.

1. The Basic Agent:
The simplest design—the "basic agent"—operates as a straightforward loop. Multiple sub-agents independently attempt to generate Lean proofs for a given problem. Each proposed proof is sent to the Lean compiler for verification. If the compiler accepts it, the problem is solved. If not, the compiler returns error messages indicating where the proof failed. The sub-agents can then use this feedback to refine their next attempts. This process repeats until a correct proof is found or a computational budget is exhausted.

International Journal of Science Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

This design is conceptually simple, and the researchers found that it successfully solved all nine Erdős problems that their more sophisticated agent solved—though at a higher computational cost for the most challenging problems.

2. The Full-Featured Agent:

The more powerful design—the "full-featured agent"—introduces two significant enhancements. First, the sub-agents are coordinated using an evolutionary algorithm, meaning that successful proof strategies are treated as "parent" solutions that combine and mutate to produce "offspring" candidates, mimicking the process of natural selection. Second, the agent can call upon AlphaProof, a specialized system for Olympiad-level Lean theorem-proving based on reinforcement learning, as a focused proof tool for particularly challenging sub-problems.

This full-featured agent is the system that achieved the study's most impressive results, autonomously solving nine Erdős problems, proving forty-four OEIS conjectures, and contributing to research across multiple fields.

V. KEY RESULTS

1. Resolving Erdős Problems:

The most striking finding of the study is that the full-featured agent autonomously resolved nine out of 353 attempted open Erdős problems. Paul Erdős, one of the most prolific mathematicians of the twentieth century, offered cash prizes for solutions to hundreds of problems spanning number theory, combinatorics, graph theory, and other fields. Many of these problems have resisted solution for decades. Among the nine problems solved by the agent were two that had remained open for 56 years.

The cost of solving each problem was remarkably low: a few hundred dollars in computational resources per problem. This efficiency suggests that AI-driven proof search is not merely a theoretical curiosity but a practical tool that can be deployed widely in mathematical research.

2. Proving OEIS Conjectures:

The Online Encyclopedia of Integer Sequences is a vast, crowd-sourced database of integer sequences, each accompanied by known properties, references, and often unsolved conjectures. The researchers tested their agent on 492 open conjectures from the OEIS. The agent successfully proved 44 of them—approximately nine percent of the attempted set. These proofs were not trivial; they represent genuine mathematical contributions that had not been previously established by human researchers.

3. Advances in Algebraic Geometry:

Beyond the structured benchmarks of Erdős problems and OEIS conjectures, the agent was deployed on original research problems across several mathematical fields. In algebraic geometry, the agent resolved a fifteen-year-old open question concerning Hilbert functions, a foundational concept in the study of polynomial rings and their geometric interpretations.

4. Improving Optimization Bounds:

In convex optimization—a field with direct applications in machine learning, engineering, and economics—the agent achieved a significant result by improving an open bound through the discovery of a novel algorithmic parameter schedule. In plain terms, the agent found a better way to set the parameters of an optimization algorithm, leading to a provably tighter bound than any previously known human-designed schedule.

International Journal of Science Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

5. Additional Contributions:

The agent also identified several misformalizations in the existing mathematical literature, meaning that it detected instances where previously published work had incorrectly formalized a problem or theorem in Lean—errors that had gone unnoticed. It contributed to resolving an open problem from Ben Green's well-known list of one hundred open questions in additive combinatorics. Furthermore, the agent is actively aiding ongoing research efforts in quantum optics and graph theory, demonstrating that this approach is not merely a one-time demonstration but a tool for continuing mathematical discovery.

6. What the Agent Design Reveals:

One of the most instructive findings of the study is a comparison between the full-featured and basic agent designs. The basic agent, which simply iterates between proof generation and Lean verification, solved all nine Erdős problems that the full-featured agent solved. The difference was not in capability but in efficiency; the full-featured agent solved the hardest problems at lower computational cost. This result is significant because it suggests that even a relatively simple design—a large language model paired with a formal verifier—may be sufficient for substantial mathematical discovery. The more sophisticated coordination mechanisms and specialized tools primarily serve to reduce cost and accelerate the search process.

7. Implications for the Future of Mathematics Research:

This study has profound implications for how mathematics may be practiced in the coming decades.

First, it demonstrates that artificial intelligence can autonomously solve open research problems—not merely competition-level exercises. The problems solved by the agent were genuine open questions, some of which had resisted human effort for over half a century. This represents a qualitative shift from previous work, which had focused on problems with known solutions.

Second, it establishes formal proof assistants as a practical interface for AI-driven mathematical discovery. By requiring that all output be written in a verifiable formal language, the approach eliminates the hallucination problem that has plagued natural-language reasoning. Any researcher can trust a Lean proof generated by the agent as much as they would trust a proof written by a human colleague—indeed, more so, because the Lean kernel has verified every step.

Third, it points to an ongoing shift from specialized, trained systems toward general-purpose agentic loops as large language models continue to improve. The basic agent design—which simply alternates generation with verification—worked. As large language models become more capable, even simpler designs may achieve comparable or superior results.

Fourth, the study highlights efficiency as a key metric for AI-driven proof search. At a cost of only a few hundred dollars per solved problem, this approach is already economically viable for many research contexts. As computational costs continue to decline, the barrier to deploying such systems will only lower.

Finally, the study raises philosophical questions about the nature of mathematical discovery. When a machine autonomously solves a problem that has eluded human mathematicians for fifty-six years, who is the discoverer? The researchers are careful to note that the agent was designed, implemented, and supervised by humans; it is a tool, not an autonomous agent in the philosophical sense. Nevertheless, the results demonstrate that tools can possess genuine discovery capabilities, and that the boundary between "assistance" and "discovery" is becoming increasingly blurred.

International Journal of Science Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

VI. CONCLUSION

The 2026 study "Advancing Mathematics Research with AI-Driven Formal Proof Search" marks a milestone in the integration of artificial intelligence into pure mathematics. By combining large language models with the Lean formal proof assistant, the researchers demonstrated that artificial intelligence can autonomously solve open research problems, including questions that had resisted human effort for over half a century. The AlphaProof Nexus framework resolved nine Erdős problems, proved forty-four OEIS conjectures, improved an optimization bound, resolved a fifteen-year-old question in algebraic geometry, and is actively contributing to ongoing research in quantum optics and graph theory. All of this was achieved at a per-problem computational cost of only a few hundred dollars.

Perhaps most significantly, the study shows that even a simple design—alternating proof generation with formal verification—is sufficient for substantial discovery. As large language models continue to improve, the capability of such systems will only grow. The era of AI-assisted—and in some cases AI-driven—mathematical research has arrived. The challenge now is not technological feasibility but cultural adoption: integrating these powerful tools into the daily practice of mathematics, while maintaining the rigor, creativity, and human insight that have always defined the discipline.

REFERENCES

1. Tsoukalas, G., Kovsharov, A., Shirobokov, S., Surina, A., Firsching, M., Bérczi, G., Ruiz, F. J. R., Suggala, A., Wagner, A. Z., Wieser, E., Yu, L., Huang, A., Horváth, M. Z., Ferrauiolo, A., Michalewski, H., Grosu, C., Hubert, T., Balog, M., Kohli, P., & Chaudhuri, S. (2026). Advancing Mathematics Research with AI-Driven Formal Proof Search. arXiv:2605.22763.
2. Tsoukalas, G. et al. (2026). AlphaProof Nexus: AI-Driven Formal Proof Search. Supplementary materials and summaries.
3. Erdős Problems Collection. Open problems in combinatorics, number theory, and graph theory.
4. Online Encyclopedia of Integer Sequences (OEIS). Database of integer sequences with conjectures and references.
5. Google DeepMind (2025). AlphaProof: Reinforcement Learning for Olympiad-Level Theorem Proving. DeepMind Publications.
6. The Lean Theorem Prover. Formal proof verification system developed at Microsoft Research.