Scipioneers Journal of Science and Technology (SJST)
‖ Website: www.scipioneers.com ‖ Email: research@scipioneers.com

Mathematical Formalization and Theorem Proving Using Artificial Intelligence: A Literature Review and Analysis of Limits

1st Ravindra P. Rewaskar
Department of Mathematics, Shankarlal Khandelwal College, Akola (M.S.), India.
Email_id: ravindrarkpr@gmail.com

Abstract:

Formalization of mathematics and automated theorem proving (ATP) have long been core topics in logic and AI. Recent years have seen a rapid infusion of statistical learning methods into proof search, premise selection, and formalization workflows; alongside this, philosophers and methodologists have examined how such systems alter mathematical practice. This review synthesizes key journal literature on (1) formalization environments and datasets, (2) machine-learning and hybrid approaches to ATP, and (3) practical and theoretical limits that remain. We concentrate on peer-reviewed journal contributions and long-form reviews to provide a firm scholarly grounding. Major conclusions are: (a) learning-assisted methods materially improve ATP performance on large formal libraries but do not eliminate fundamental logical limits; (b) hybrid architectures (learning + symbolic verification) are currently the most promising route to useful automation; and (c) concerns about data scarcity, interpretability, computational cost, evaluation methodology, and epistemic questions for mathematics persist and require coordinated technical and philosophical work.

Keywords: ATP, Hybrid Architecture, Formalization.

I INTRODUCTION

Mathematics formalization—the process of translating informal mathematical reasoning into machine-checkable formal representations—underpins modern interactive theorem provers (ITPs) and automated theorem provers (ATPs). Historically, ATPs were predominantly symbolic search engines with carefully engineered heuristics; from roughly the 2010s onward, machine learning has been introduced to guide search, rank premises, and suggest proof steps, yielding measurable empirical gains on large formal corpora [1–4]. This article reviews peer-reviewed, journal-level work and long-form reviews that document those advances and reflect on their limits.

II. FORMALIZATION INFRASTRUCTURES AND EMPIRICAL DATASETS

Large formal libraries (Mizar, Flyspeck, HOL Light, and others) supply the corpora on which learning methods are trained and evaluated. Journal publications have documented how machine-assisted workflows exploit those libraries: Kaliszyk & Urban describe how Flyspeck was coupled to external ATPs and machine-learned premise selection methods to produce push-button proofs for a substantial fraction of Flyspeck theorems [1]. Follow-on work formalized strategies for mining and re-using millions of derived lemmas to enlarge the effective search space while keeping retrieval tractable [3]. These efforts established data-driven workflows (export dependency information, represent formulas for learning, integrate learned advisors into ITPs) that underpin nearly all subsequent ML+ATP work [1,3]. The existence, scale, and provenance of formal corpora therefore shape what learning methods can achieve in ATP; this connection has been documented in several journal papers and reviews [1–4].

III. FAMILIES OF LEARNING-BASED METHODS IN ATP:

Research articles, available literature and long reviews identify three broad families of approaches.

3.1 Learning for premise selection and retrieval.

Premise selection—choosing a small, relevant subset of axioms/lemmas from a very large library—is perhaps the most mature ML application in ATP. Kernel- and corpus-analysis methods, and later neural embedding/graph-based encoders, have been shown (in journal reports) to increase ATP success rates by making the search tractable in large theories [2,3]. Representative journal-level evaluations demonstrate that learned filters substantially increase the proportion of theorems an ATP can solve in constrained time budgets [2,3].

3.2 Learning to guide symbolic search.

Rather than replace symbolic proof engines, many systems use learned models (logistic/forest classifiers, neural nets, GNNs) to guide which inference rule, clause, or tactic to apply next. Journal articles and detailed surveys show that integrating learned guidance into symbolic search workflows improves performance across benchmarks derived from real formal libraries [1–4].

3.3 Hybrid and verification-first architectures.

A prevailing pattern in the literature is the hybrid pipeline: a statistical component proposes candidate steps or premises and a symbolic checker (the prover/ITP kernel) verifies correctness. This separation—statistical generation followed by formal verification—preserves mathematical standards while allowing machine learning to propose creative or nontrivial steps. Long reviews summarize this architecture as the pragmatic path forward for robust automation [4].

IV. MEASURED PROGRESS AND REPRESENTATIVE RESULTS

Journal reports and monographic reviews present concrete empirical gains:

Learning-assisted pipelines enabled automatic proofs of nontrivial fractions of large libraries: Kaliszyk & Urban report substantial automated proof coverage for Flyspeck when premise selection and ATP combination are used [1].

Improvements in lemma mining and reuse show that expanding the set of “usable” lemmas—chosen by learned usefulness measures—boosts automatic proving on subsequent conjectures [3].

Long-form, peer-reviewed reviews summarize the progress of learning-guided SAT/QSAT and ATP integration and document that ML methods substantially improve solver heuristics in practice while also highlighting methodological caveats (overfitting, dataset contamination) [4].

These journal studies supply the strongest empirical evidence that ML meaningfully improves ATP performance on existing formal corpora, while also documenting the need for careful evaluation protocols. [1,3,4]

V. PRACTICAL AND CONCEPTUAL LIMITS

Despite clear progress, peer-reviewed work and methodological reviews identify persistent limitations and principled boundaries.

5.1 Data limitations and distributional fragility.

Formalized mathematics corpora—even Flyspeck or Mizar—are small relative to natural-language corpora used to train general LLMs; many theorems and proof patterns are rare, which challenges generalization. Journal analyses show that premise-selection and lemma-mining methods help but cannot fully substitute for deep, varied training distributions; performance often degrades on novel problem distributions or on theorems substantially different from training examples [1–4].

5.2 Correctness, verification, and brittleness.

Statistical generators can propose plausible but incorrect steps. The hybrid pattern ensures that only formally correct proofs are admitted, but the conversion of generated output into machine-checkable objects is error-prone and can mask subtle failures of the generator (typos, type mismatches, hidden assumptions). Journal accounts emphasize that the separation of generation and verification is necessary but not sufficient: robust pipelines must also handle partial proofs, tactic failures, and representation mismatches [1,4].

5.3 Interpretability and mathematical value.

Philosophical and methodological journal articles discuss whether machine-found proofs provide human-accessible mathematical insight. Even when an ATP produces a valid proof, it may be combinatorial, unilluminating, or rely on many small lemmas rather than the concise, conceptually enlightening arguments mathematicians prefer. Work in philosophy of science and computational epistemology highlights this mismatch between formal correctness and mathematical value, arguing that acceptance depends on the mathematical community's willingness to treat machine-generated output as explanatory evidence [5,6].

5.4 Computational and reproducibility constraints.

State-of-the-art learned guidance often requires substantial compute (training encoders, running heavy search). Journal reviews of ML for SAT/QSAT and ATP note the practical barrier this poses to reproducibility and broad adoption; resource-efficient variants and shared benchmarks are advocated as remedies [4].

5.5 Theoretical/Foundational limits.

No ML system can bypass formal limits set by logic and computability: Gödel's incompleteness and undecidability results imply that within any sufficiently expressive formal system some true statements are unprovable; complexity theoretic hardness bounds also limit automatability in general. Journal philosophical treatments emphasize that ML-assisted proving operates inside these formal constraints and cannot "defeat" them—what it can do is make particular classes of proofs easier to find or verify [5].

VI. DIRECTIONS RECOMMENDED BY JOURNAL LITERATURE:

Peer-reviewed sources converge on several avenues likely to yield meaningful progress:

1. Hybrid, verifiable systems. Continue coupling statistical proposal mechanisms with strong symbolic kernels to maintain correctness guarantees [1,4].
2. Better representations. Journal research on graph and semantic encodings of formulas suggests improved generalization and premise retrieval performance if representations capture variable renaming invariances and deeper semantic relations [2,3].
3. Data curation and augmentation. Systematic lemma mining, synthetic proof generation (with verification), and carefully curated benchmarks are recommended to reduce scarcity and distributional gaps [3].
4. Evaluation reform. Adopt contamination controls, richer metrics (novelty, concision, human acceptance) and shared reproducible benchmarks [1,4].
5. Interdisciplinary scrutiny. Philosophy and methodology journals recommend integrating sociotechnical analysis—examining how mathematicians will adopt, inspect, and accept machine-produced proofs—into technical research agendas [5,6].

VII. CONCLUSION

By enhancing premise selection, lemma reuse, and heuristic guiding for symbolic engines, machine learning into ATP pipelines significantly boosts performance on big, formal datasets. However, there are still technical issues (such as data shortages, fragility and computational expense); scientific literature emphasizes that advancement will be gradual and that independent verification and meticulous evaluation are necessary to maintain mathematical standards. Researchers must integrate enhanced models, reliable testing processes, deeper illustrations, and community-focused review procedures if the area is to go from striking results from experimentation to widespread mathematics influence.

VIII. REFERENCES

1. C. Kaliszyk and J. Urban, "Learning-Assisted Automated Reasoning with Flyspeck," Journal of Automated Reasoning, vol. 53, pp. 173–213, 2014. DOI: 10.1007/s10817-014-9303-3
2. J. Alama, T. Heskes, D. Kühlwein, E. Tsivtsivadze, and J. Urban, "Premise selection for mathematics by corpus analysis and kernel methods," Journal of Automated Reasoning, vol. 52, no. 2, pp. 191–213, 2014.
3. C. Kaliszyk and J. Urban, "Learning-assisted theorem proving with millions of lemmas," Journal of Symbolic Computation, vol. 69, pp. 109–128, 2015.
4. S. B. Holden, Machine Learning for Automated Theorem Proving: Learning to Solve SAT and QSAT, Foundations and Trends® in Machine Learning, vol. 14, no. 6, pp. 807–989, 2021.
5. M. Pantsar, "Theorem proving in artificial neural networks: new frontiers in mathematical AI," European Journal for Philosophy of Science, 2024.
6. S. Almpani, P. Stefaneas, and I. Vandoulakis, "Formalization of mathematical proof practice through an argumentation-based model," Axiomathes (Global Philosophy / Axiomathes series), vol. 33, no. 3, pp. 1–28, 2023. DOI: 10.1007/s10516-023-09685-z.
7. Blanchette, J. C., M. Fleury, P. Lammich, and C. Weidenbach, "A Verified SAT Solver Framework with Learn, Forget, Restart and Incrementality". Journal of Automated Reasoning. 61(1–5): 333–365, 2018.
8. Pantsar, M., "Theorem proving in artificial neural networks: new frontiers in mathematical AI." European Journal for Philosophy of Science, 14(1), Article 4, 2024 https://doi.org/10.1007/s13194-024-00569-6