Novelty Assessment Report

Paper: ProofOptimizer: Training Language Models to Simplify Proofs without Human Demonstrations
PDF URL: https://openreview.net/pdf?id=huptrb4JTa
Venue: ICLR 2026 Conference Submission
Year: 2026
Report Generated: 2026-01-04

Abstract

Neural theorem proving has advanced rapidly in the past year, reaching IMO gold-medalist capabilities and producing formal proofs that span thousands of lines. Although such proofs are mechanically verified by formal systems like Lean, their excessive length renders them difficult for humans to comprehend and limits their usefulness for mathematical insight. Proof simplification is therefore a critical bottleneck. Yet, training data for this task is scarce, and existing methods—mainly agentic scaffolding with off-the-shelf LLMs—struggle with the extremely long proofs generated by RL-trained provers. We introduce ProofOptimizer, the first language model trained to simplify Lean proofs without requiring additional human supervision. ProofOptimizer is trained via expert iteration and reinforcement learning, using Lean to verify simplifications and provide training signal. At inference time, it operates within an iterative proof-shortening workflow, progressively reducing proof length. Experiments show that ProofOptimizer substantially compresses proofs generated by state-of-the-art RL-trained provers on standard benchmarks, reducing proof length by 87% on miniF2F, 57% on PutnamBench, and 50% on Seed-Prover's IMO 2025 proofs. Beyond conciseness, the simplified proofs check faster in Lean and further improve downstream prover performance when reused as training data for supervised finetuning.

Core Task Landscape

This paper addresses: **Simplifying Formal Mathematical Proofs Using Language Models**

A total of **50 papers** were analyzed and organized into a taxonomy with **31 categories**.

Taxonomy Overview

The research landscape has been organized into the following main categories:

- **Formal Proof Generation and Verification**
- **Autoformalization and Translation**
- **Proof Optimization and Repair**
- **Specialized Mathematical Domains**
- **Informal Mathematical Reasoning**
- **Benchmarks and Datasets**
- **Foundation Models and Pretraining**
- **Software Verification Applications**
- **Surveys and Theoretical Foundations**

Complete Taxonomy Tree

- Simplifying Formal Mathematical Proofs Using Language Models Survey Taxonomy
- Formal Proof Generation and Verification
  - Whole-Proof Generation Approaches (2 papers)
  - [20] Baldur: Whole-proof generation and repair with large language models (Emily First, 2023) View paper
  - [30] Neural Theorem Proving: Generating and Structuring Proofs for Formal Verification (Eiers, 2025) View paper
  - Tactic-Based Proof Construction (3 papers)
  - [8] Proof Automation with Large Language Models (Minghai Lu, 2024) View paper
  - [17] Generative language modeling for automated theorem proving (Polu, 2020) View paper
  - [41] Proof artifact co-training for theorem proving with language models (Han, 2021) View paper
  - Retrieval-Augmented Theorem Proving (2 papers)
  - [9] MLFMF: data sets for machine learning for mathematical formalization (Bauer, 2023) View paper
  - [10] Leandojo: Theorem proving with retrieval-augmented language models (Yang Kaiyu, 2023) View paper
  - Neuro-Symbolic Proof Systems (2 papers)
  - [15] ProofNet++: A Neuro-Symbolic System for Formal Proof Verification with Self-Correction (Ambati, 2025) View paper
  - [33] Proving Olympiad Inequalities by Synergizing LLMs and Symbolic Reasoning (Li Zenan, 2025) View paper
  - Agent-Based Proof Frameworks (3 papers)
  - [24] APOLLO: Automated LLM and Lean Collaboration for Advanced Formal Reasoning (Farnia, 2025) View paper
  - [35] HybridProver: Augmenting Theorem Proving with LLM-Driven Proof Synthesis and Refinement (Hu Jilin, 2025) View paper
  - [50] Prover Agent: An Agent-based Framework for Formal Mathematical Proofs (Liu Chaoran, 2025) View paper
  - Recursive and Decomposition-Based Proving (2 papers)
  - [27] Solving formal math problems by decomposition and iterative reflection (Zhou Yichi, 2025) View paper
  - [32] Hilbert: Recursively Building Formal Proofs with Informal Reasoning (Varambally, 2025) View paper
- Autoformalization and Translation
  - Statement-Level Autoformalization (2 papers)
  - [14] Towards a Mathematics Formalisation Assistant using Large Language Models (Agrawal, 2022) View paper

    - [18] Autoformalize mathematical statements by symbolic equivalence and semantic consistency (Zenan Li, 2024) View paper
    - Proof-Level Autoformalization (1 papers)
    - [31] Don't Trust: Verify--Grounding LLM Quantitative Reasoning with Autoformalization (Zhou, 2024) View paper
    - Multilingual Autoformalization (1 papers)
    - [25] Towards multilingual autoformalization and informalization of mathematics (Ranta, 2024) View paper
    - Verification-Oriented Autoformalization (2 papers)
    - [19] From Informal to Formal--Incorporating and Evaluating LLMs on Natural Language Requirements to Verifiable Formal Proofs (Jialun Cao, 2025) View paper
    - [29] PAT-Agent: Autoformalization for Model Checking (Zuo Xinyue, 2025) View paper
- Proof Optimization and Repair
    - Proof Simplification and Compression ★ (2 papers)
    - [0] ProofOptimizer: Training Language Models to Simplify Proofs without Human Demonstrations (Anon et al., 2026) View paper
    - [7] FVEL: Interactive formal verification environment with large language models via theorem proving (Qingxing Cao, 2024) View paper
    - Proof Repair and Error Correction (1 papers)
    - [28] Faithful and robust llm-driven theorem proving for nli explanations (Quan-xin, 2025) View paper
- Specialized Mathematical Domains
    - Inequality Proving (1 papers)
    - [4] Solving Inequality Proofs with Large Language Models (Sheng Jia-Yi, 2025) View paper
    - Mathematical Analysis (1 papers)
    - [6] Large Language Models for Mathematical Analysis (Chen Ziye, 2024) View paper
    - Trigonometric Expression Reduction (1 papers)
    - [38] Trigo: Benchmarking formal mathematical proof reduction for generative language models (Cao, 2023) View paper
- Informal Mathematical Reasoning
    - Chain-of-Thought Mathematical Reasoning (2 papers)
    - [2] Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought (Saparov, 2022) View paper
    - [5] A survey on large language models for mathematical reasoning (Wang Peng-Yuan, 2025) View paper
    - Verification-Enhanced Informal Reasoning (1 papers)
    - [21] Safe: Enhancing Mathematical Reasoning in Large Language Models via Retrospective Step-aware Formal Verification (Cheng-wu Liu, 2025) View paper
    - Mathematical Rule Learning (1 papers)
    - [44] Learning Mathematical Rules with Large Language Models (Antoine Gorceix, 2024) View paper
    - Educational Mathematical Reasoning (1 papers)
    - [11] â large language models to support pre-service teachers mathematical reasoningâan exploratory study on ChatGPT as an instrument for creating mathematical proofs â (F Dilling, 2024) View paper
- Benchmarks and Datasets
    - Formal Proof Benchmarks (1 papers)
    - [13] Formalmath: Benchmarking formal mathematical reasoning of large language models (Yu, 2025) View paper
    - Human-Evaluated Proof Collections (1 papers)
    - [49] The Open Proof Corpus: A Large-Scale Study of LLM-Generated Mathematical Proofs (Petrov Ivo, 2025) View paper
    - Informal Reasoning Benchmarks (1 papers)
    - [1] Large language models for mathematical reasoning: Progresses and challenges (Janice Ahn, 2024) View paper
- Foundation Models and Pretraining
    - Mathematics-Specialized Language Models (1 papers)
    - [3] Llemma: An open language model for mathematics (Azerbayev, 2023) View paper
    - Formal Logic Learning (1 papers)
    - [39] Can Large Language Models Learn Formal Logic? A Data-Driven Training and Evaluation Framework (xia Yuan, 2025) View paper
    - Interpretability and Representation Control (1 papers)
    - [46] Localist LLMs - A Mathematical Framework for Dynamic Locality Control (Diederich, 2025) View paper
- Software Verification Applications
    - Specification Generation from Requirements (2 papers)
    - [34] SLD-Spec: Enhancement LLM-assisted Specification Generation for Complex Loop Functions via Program Slicing and Logical Deletion (Chen, 2025) View paper
    - [42] Supporting Software Formal Verification with Large Language Models: An Experimental Study (Wang Wei-qi, 2025) View paper
    - Code Verification and Bug Detection (2 papers)
    - [36] PyVeritas: On Verifying Python via LLM-Based Transpilation and Bounded Model Checking for C (Orvalho, 2025) View paper
    - [47] LLM-assisted Bug Identification and Correction for Verilog HDL (Khushboo Qayyum, 2025) View paper
- Surveys and Theoretical Foundations
    - General Mathematical Reasoning Surveys (2 papers)
    - [23] A survey on mathematical reasoning and optimization with large language models (Forootani, 2025) View paper
    - [26] Thinking machines: Mathematical reasoning in the age of llms (Asperti, 2025) View paper
    - Theorem Proving Surveys (1 papers)
    - [12] A survey on deep learning for theorem proving (Li Zhaoyu, 2024) View paper
    - Autoformalization Surveys (1 papers)
    - [45] Autoformalization in the Era of Large Language Models: A Survey (Weng Ke, 2025) View paper
    - Theoretical Foundations and Position Papers (6 papers)
    - [16] Formal mathematical reasoning: A new frontier in ai (Yang Kaiyu, 2024) View paper
    - [22] AI for Mathematics (Qinghai Miao, 2024) View paper
    - [37] Machine assisted proof (Tao, 2024) View paper
    - [40] Mathematics in the Age of Large Language Models (Chojecki, 2025) View paper
    - [43] Automating mathematical proof generation using large language model agents and knowledge graphs (Li, 2025) View paper

- [48] A Natural Formalized Proof Language (Lihan Xie, 2024) View paper

Narrative

Core task: Simplifying formal mathematical proofs using language models. The field has evolved into a rich ecosystem of interconnected branches. Formal Proof Generation and Verification focuses on end-to-end theorem proving systems that synthesize and check proofs in systems like Lean or Coq, with works such as LeanDojo[10] and Deep Learning Theorem Proving[12] establishing foundational infrastructure. Autoformalization and Translation addresses the challenge of converting informal mathematical statements into formal syntax, exemplified by efforts like Autoformalize[18] and Multilingual Autoformalization[25]. Proof Optimization and Repair targets the refinement of existing proofs—making them shorter, more readable, or correcting errors—while Specialized Mathematical Domains tackle specific problem classes such as inequalities or trigonometry. Foundation Models and Pretraining, including Llemma[3], provide the base capabilities that other branches build upon, and Benchmarks and Datasets supply the evaluation infrastructure that drives progress across all areas.

Within Proof Optimization and Repair, a small but growing cluster of works explores proof simplification and compression, seeking to reduce proof complexity while preserving correctness. ProofOptimizer[0] sits squarely in this niche, emphasizing automated techniques to streamline formal proofs. Nearby, FVEL[7] addresses related verification and efficiency concerns, though with a slightly different emphasis on validation workflows. This contrasts with broader proof generation efforts like Proof Automation[8] or MLFMF[9], which prioritize discovering new proofs over refining existing ones. The tension between generating correct proofs and making them human-readable or computationally efficient remains a central open question. ProofOptimizer[0] contributes to this dialogue by focusing specifically on simplification, complementing the wider landscape where most attention has centered on proof discovery and autoformalization rather than post-hoc optimization.

Related Works in Same Category

The following **1 sibling papers** share the same taxonomy leaf node with the original paper:

1. FVEL: Interactive formal verification environment with large language models via theorem proving

Authors: Qingxing Cao, Yinya Huang, Xiaodan Liang, Xiaohan Lin, Liu Zhengying, et al. (8 authors total) | **Year/Venue**: 2024 | **URL**: View paper

Abstract

â¦ can further simplify and abstract the generated SIMPL language, producing a higher-level functional specification that is easier to reason by humans. We provide the simplified Isabelle â¦

Relationship Analysis

Both papers belong to the Proof Simplification and Compression category, focusing on reducing proof complexity while maintaining correctness. However, ProofOptimizer specifically trains language models to simplify existing Lean proofs generated by RL-trained provers using expert iteration and reinforcement learning, achieving significant length reductions (87% on miniF2F). In contrast, FVEL focuses on interactive formal verification of C code by transforming programs into Isabelle and conducting theorem proving with LLMs, rather than simplifying already-generated proofs.

Contributions Analysis

Overall novelty summary. ProofOptimizer introduces the first language model trained specifically to simplify Lean proofs without additional human supervision, using expert iteration and reinforcement learning with Lean-based verification as the training signal. The paper sits in the 'Proof Simplification and Compression' leaf under 'Proof Optimization and Repair', which contains only two papers total in the entire taxonomy. This represents a notably sparse research direction within the broader field of 50 papers, suggesting that proof optimization has received far less attention than proof generation or autoformalization tasks.

The taxonomy reveals that most research effort concentrates in adjacent branches: 'Formal Proof Generation and Verification' contains multiple dense subtopics with 15 papers across six leaves, while 'Autoformalization and Translation' addresses informal-to-formal conversion with seven papers. The 'Proof Repair and Error Correction' sibling leaf focuses on fixing incorrect proofs rather than simplifying correct ones. ProofOptimizer's work diverges from these neighboring directions by assuming correct input proofs and targeting length reduction, rather than initial generation, translation, or error correction.

Among 30 candidates examined across three contributions, none were identified as clearly refuting the paper's claims. The first contribution (ProofOptimizer as first trained simplification model) examined 10 candidates with zero refutable matches. The training methodology and iterative inference workflow contributions each examined 10 candidates with similar results. This suggests that within the limited search scope, no prior work directly addresses supervised learning for proof simplification in Lean, though the small candidate pool means the search cannot be considered exhaustive.

Based on the limited literature search covering 30 semantically similar papers, ProofOptimizer appears to occupy a genuinely sparse research area. The taxonomy structure confirms that proof optimization receives minimal attention compared to proof generation. However, the analysis cannot rule out relevant work outside the top-30 semantic matches or in adjacent communities not captured by this search methodology.

This paper presents **3 main contributions**, each analyzed against relevant prior work:

Contribution 1: ProofOptimizer: first language model trained to simplify Lean proofs without human supervision

Description: The authors present ProofOptimizer, a language model specifically trained for proof simplification in Lean using expert iteration and reinforcement learning, without needing human-annotated simplification data. The model uses Lean's verification to provide training signals and operates within an iterative proof-shortening workflow at inference time.

This contribution was assessed against **10 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

1. Baldur: Whole-proof generation and repair with large language models

URL: View paper

Brief Assessment

Baldur[20] focuses on whole-proof generation and repair for Isabelle/HOL, not proof simplification in Lean. The tasks and systems are fundamentally different.

---

2. Kimina-prover preview: Towards large formal reasoning models with reinforcement learning

URL: View paper

Brief Assessment

Kimina-Prover[71] focuses on proof generation using reinforcement learning with a formal reasoning pattern, not on proof simplification. The two systems address different tasks in the theorem proving pipeline.

---

3. Automating mathematical proof generation using large language model agents and knowledge graphs

URL: View paper

Brief Assessment

Automating Mathematical Proof[43] focuses on automated theorem proving using knowledge graphs and natural language proof generation, not on training models to simplify existing formal proofs. The candidate does not address proof simplification as a training objective.

---

4. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs

URL: View paper

Brief Assessment

Draft Sketch Prove[70] focuses on translating informal proofs to formal proof sketches in Isabelle, not on training models to simplify existing formal proofs in Lean without human supervision.

---

5. APOLLO: Automated LLM and Lean Collaboration for Advanced Formal Reasoning

URL: View paper

Brief Assessment

APOLLO[24] focuses on proof repair and error correction using compiler feedback, not proof simplification or shortening. The two systems address different problems in automated theorem proving.

---

6. Generative language modeling for automated theorem proving

URL: View paper

Brief Assessment

Generative Language Modeling[17] focuses on automated theorem proving in Metamath, not proof simplification in Lean. The candidate demonstrates proof generation capabilities but does not address the specific task of training models to simplify existing proofs without human supervision.

---

7. Position: Formal Mathematical Reasoningâ□□A New Frontier in AI

URL: View paper

Brief Assessment

Formal Mathematical Reasoning[74] is a position paper discussing the broader landscape of formal mathematical reasoning and AI. It does not present a specific system for proof simplification trained without human supervision, focusing instead on surveying the field and proposing future directions.

---

8. Autoformalization with large language models

URL: View paper

Brief Assessment

Autoformalization LLMs[72] focuses on translating natural language mathematics to formal specifications (autoformalization), not on simplifying existing formal proofs. The candidate addresses a different task in the formal mathematics pipeline.

---

9. Towards automating formalisation of theorem statements using large language models

URL: View paper

Brief Assessment

Automating Formalisation[75] focuses on translating natural language theorem statements into Lean formal statements (autoformalisation), not on simplifying existing formal proofs. The tasks are fundamentally different.

---

10. Formal theorem proving by rewarding llms to decompose proofs hierarchically

URL: View paper

Brief Assessment

Hierarchical Decomposition[73] focuses on decomposing proofs hierarchically using reinforcement learning to propose and prove lemmas in Isabelle, not on simplifying existing proofs in Lean without human supervision.

Contribution 2: Training methodology combining expert iteration and reinforcement learning for proof simplification

Description: The authors develop a training approach that combines expert iteration (where the model proposes simplifications verified by Lean and incorporated into training data) and online reinforcement learning (using proof length and correctness as reward signals) to enable continual improvement in proof simplification.

This contribution was assessed against **10 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

1. Formal mathematics statement curriculum learning

URL: View paper

Brief Assessment

Statement Curriculum Learning[54] focuses on expert iteration for theorem proving (generating proofs from statements), not proof simplification. The candidate applies expert iteration to solve increasingly difficult formal mathematics problems, while the original paper uses it to simplify existing proofs.

---

2. Goedel-Prover-V2: Scaling Formal Theorem Proving with Scaffolded Data Synthesis and Self-Correction

URL: View paper

Brief Assessment

Goedel-Prover[59] focuses on formal theorem proving (generating proofs from statements) rather than proof simplification. While both use expert iteration and RL, the tasks differ fundamentally: the candidate generates proofs, whereas the original simplifies existing proofs.

---

3. GAR: Generative Adversarial Reinforcement Learning for Formal Theorem Proving

URL: View paper

Brief Assessment

GAR[58] focuses on adversarial training between a problem composer and solver for theorem proving, not on proof simplification. The candidate's methodology trains models to generate harder problems and solve them, whereas the original trains models to simplify existing proofs.

---

4. Bfs-prover: Scalable best-first tree search for llm-based automatic theorem proving

URL: View paper

Brief Assessment

BFS-Prover[55] focuses on best-first tree search for theorem proving, not proof simplification. While both use expert iteration, BFS-Prover applies it to improve proof search capabilities, whereas the original work targets proof length reduction.

---

5. Internlm2. 5-stepprover: Advancing automated theorem proving via expert iteration on large-scale lean problems

URL: View paper

Brief Assessment

InternLM StepProver[53] focuses on theorem proving with critic-guided search for tactic generation, not proof simplification. The candidate uses expert iteration to improve a prover-critic framework for discovering proofs, while the original paper trains models specifically to simplify existing proofs by reducing their length.

---

6. Lean-star: Learning to interleave thinking and proving

URL: View paper

Brief Assessment

Lean-star[52] focuses on theorem proving (generating proofs from scratch) rather than proof simplification. The expert iteration in Lean-star[52] is used to improve proof generation capabilities, not to simplify existing proofs.

---

7. AI for Mathematics

URL: View paper

Brief Assessment

AI for Mathematics[22] discusses mathematical exploration and proof verification broadly, but does not present a specific training methodology combining expert iteration and RL for proof simplification tasks.

---

8. Contributions to Neural Theorem Proving

URL: View paper

Brief Assessment

Neural Theorem Contributions[56] focuses on data synthesis and expert iteration for theorem proving in general, not specifically for proof simplification. The candidate addresses data scarcity for high-level reasoning steps, while the original paper targets proof simplification as a distinct task with its own training methodology.

---

9. STP: Self-play LLM Theorem Provers with Iterative Conjecturing and Proving

URL: View paper

Brief Assessment

STP[57] focuses on self-play between conjecturer and prover roles for theorem proving, not on proof simplification. The training methodology in STP[57] involves generating new conjectures and proving them, which is fundamentally different from the original paper's focus on simplifying existing proofs.

---

10. ABEL: Sample efficient online reinforcement learning for neural theorem proving

URL: View paper

Brief Assessment

ABEL[51] focuses on theorem proving (finding proofs), not proof simplification (shortening existing proofs). The candidate uses expert iteration and RL for proof search, while the original applies these techniques to the distinct task of simplifying already-generated proofs.

Contribution 3: Iterative proof-shortening inference workflow

Description: The authors introduce an inference-time algorithm that iteratively applies the model to progressively shorten proofs by sampling multiple candidate simplifications and repeatedly applying the model to the currently shortest proof, achieving substantial compression on benchmark datasets.

This contribution was assessed against **10 related papers** from the literature. Papers with potential prior art are analyzed in detail with textual evidence; others receive brief assessments.

---

1. Efficient, verified checking of propositional proofs

URL: View paper

Brief Assessment

Verified Checking Proofs[68] focuses on verifying the correctness of SAT proof checkers through formal verification in ACL2, not on iteratively shortening proofs. The paper develops increasingly efficient verified checkers but does not address proof simplification or iterative shortening workflows.

---

2. Stepwise refinement provenance scheme for wireless sensor networks

URL: View paper

Brief Assessment

Stepwise Refinement Provenance[66] addresses data provenance encoding in wireless sensor networks, not proof simplification or theorem proving. The domains are entirely different (network protocols vs. formal mathematics).

---

3. Learning better representations from less data for propositional satisfiability

URL: View paper

Brief Assessment

Propositional Satisfiability Representations[62] focuses on resolution proofs for SAT problems using attention mechanisms, not general proof simplification workflows. The iterative approach in the candidate involves bootstrapped training where model-generated proofs replace teacher proofs, which is domain-specific to propositional logic rather than the general theorem proving context of the original paper.

---

4. How to discover short, shorter, and the shortest proofs of unsatisfiability: a branch-and-bound approach for resolution proof length minimization

URL: View paper

Brief Assessment

Resolution Proof Minimization[61] focuses on finding the shortest resolution proofs for unsatisfiable formulas using a branch-and-bound approach with layer list representations. This is fundamentally different from the original paper's iterative inference-time workflow that progressively shortens Lean proofs by repeatedly applying a language model to sample candidate simplifications.

---

5. Toward mechanical mathematics

URL: View paper

Brief Assessment

Mechanical Mathematics[69] focuses on automated theorem proving in formal logic systems (propositional and predicate calculus) from 1960, not on iteratively shortening already-generated proofs. The candidate discusses proof procedures and decision methods, while the original paper introduces a modern inference-time algorithm for progressively compressing proofs generated by RL-trained language models.

---

6. Concise: Confidence-guided compression in step-by-step efficient reasoning

URL: View paper

Brief Assessment

CONCISE[60] focuses on compressing step-by-step reasoning chains in large reasoning models through confidence-guided mechanisms during generation, not on iteratively shortening formal proofs in theorem proving systems like Lean.

---

7. The normalized curve shortening flow and homothetic solutions

URL: View paper

Brief Assessment

Normalized Curve Shortening[64] addresses differential geometry problems involving curve evolution equations, not proof simplification or iterative workflows for reducing proof length in formal theorem proving systems.

---

8. Coarsening natural deduction proofs II: finding gaunt proofs

URL: View paper

Brief Assessment

Coarsening Natural Deduction[65] focuses on inductive transformations of natural deduction proofs to remove irrelevancies in formal logic systems (core logic and classical core logic), not on iterative inference-time algorithms for progressively shortening proofs through repeated model application and sampling.

---

9. Abstracting gradual typing

URL: View paper

Brief Assessment

Abstracting Gradual Typing[63] focuses on gradual type systems and abstract interpretation for programming languages, not proof simplification or iterative workflows for reducing proof length in theorem proving systems.

---

10. Rodin: an open toolset for modelling and reasoning in Event-B

URL: View paper

Brief Assessment

Rodin[67] focuses on event-B modeling and theorem proving with proof tree construction and management, not on iterative proof-shortening workflows that progressively reduce proof length through repeated sampling and simplification.

Appendix: Text Similarity Detection

No high-similarity text segments were detected across any compared papers.

References

- [0] ProofOptimizer: Training Language Models to Simplify Proofs without Human Demonstrations View paper
- [1] Large language models for mathematical reasoning: Progresses and challenges View paper
- [2] Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-Thought View paper
- [3] Llemma: An open language model for mathematics View paper
- [4] Solving Inequality Proofs with Large Language Models View paper
- [5] A survey on large language models for mathematical reasoning View paper
- [6] Large Language Models for Mathematical Analysis View paper
- [7] FVEL: Interactive formal verification environment with large language models via theorem proving View paper
- [8] Proof Automation with Large Language Models View paper
- [9] MLFMF: data sets for machine learning for mathematical formalization View paper
- [10] Leandojo: Theorem proving with retrieval-augmented language models View paper
- [11] â large language models to support pre-service teachers mathematical reasoningâan exploratory study on ChatGPT as an instrument for creating mathematical proofs â View paper
- [12] A survey on deep learning for theorem proving View paper
- [13] Formalmath: Benchmarking formal mathematical reasoning of large language models View paper
- [14] Towards a Mathematics Formalisation Assistant using Large Language Models View paper
- [15] ProofNet++: A Neuro-Symbolic System for Formal Proof Verification with Self-Correction View paper
- [16] Formal mathematical reasoning: A new frontier in ai View paper
- [17] Generative language modeling for automated theorem proving View paper
- [18] Autoformalize mathematical statements by symbolic equivalence and semantic consistency View paper
- [19] From Informal to Formal--Incorporating and Evaluating LLMs on Natural Language Requirements to Verifiable Formal Proofs View paper
- [20] Baldur: Whole-proof generation and repair with large language models View paper
- [21] Safe: Enhancing Mathematical Reasoning in Large Language Models via Retrospective Step-aware Formal Verification View paper

- [22] AI for Mathematics View paper
- [23] A survey on mathematical reasoning and optimization with large language models View paper
- [24] APOLLO: Automated LLM and Lean Collaboration for Advanced Formal Reasoning View paper
- [25] Towards multilingual autoformalization and informalization of mathematics View paper
- [26] Thinking machines: Mathematical reasoning in the age of llms View paper
- [27] Solving formal math problems by decomposition and iterative reflection View paper
- [28] Faithful and robust llm-driven theorem proving for nli explanations View paper
- [29] PAT-Agent: Autoformalization for Model Checking View paper
- [30] Neural Theorem Proving: Generating and Structuring Proofs for Formal Verification View paper
- [31] Don't Trust: Verify--Grounding LLM Quantitative Reasoning with Autoformalization View paper
- [32] Hilbert: Recursively Building Formal Proofs with Informal Reasoning View paper
- [33] Proving Olympiad Inequalities by Synergizing LLMs and Symbolic Reasoning View paper
- [34] SLD-Spec: Enhancement LLM-assisted Specification Generation for Complex Loop Functions via Program Slicing and Logical Deletion View paper
- [35] HybridProver: Augmenting Theorem Proving with LLM-Driven Proof Synthesis and Refinement View paper
- [36] PyVeritas: On Verifying Python via LLM-Based Transpilation and Bounded Model Checking for C View paper
- [37] Machine assisted proof View paper
- [38] Trigo: Benchmarking formal mathematical proof reduction for generative language models View paper
- [39] Can Large Language Models Learn Formal Logic? A Data-Driven Training and Evaluation Framework View paper
- [40] Mathematics in the Age of Large Language Models View paper
- [41] Proof artifact co-training for theorem proving with language models View paper
- [42] Supporting Software Formal Verification with Large Language Models: An Experimental Study View paper
- [43] Automating mathematical proof generation using large language model agents and knowledge graphs View paper
- [44] Learning Mathematical Rules with Large Language Models View paper
- [45] Autoformalization in the Era of Large Language Models: A Survey View paper
- [46] Localist LLMs - A Mathematical Framework for Dynamic Locality Control View paper
- [47] LLM-assisted Bug Identification and Correction for Verilog HDL View paper
- [48] A Natural Formalized Proof Language View paper
- [49] The Open Proof Corpus: A Large-Scale Study of LLM-Generated Mathematical Proofs View paper
- [50] Prover Agent: An Agent-based Framework for Formal Mathematical Proofs View paper
- [51] ABEL: Sample efficient online reinforcement learning for neural theorem proving View paper
- [52] Lean-star: Learning to interleave thinking and proving View paper
- [53] Internlm2. 5-stepprover: Advancing automated theorem proving via expert iteration on large-scale lean problems View paper
- [54] Formal mathematics statement curriculum learning View paper
- [55] Bfs-prover: Scalable best-first tree search for llm-based automatic theorem proving View paper
- [56] Contributions to Neural Theorem Proving View paper
- [57] STP: Self-play LLM Theorem Provers with Iterative Conjecturing and Proving View paper
- [58] GAR: Generative Adversarial Reinforcement Learning for Formal Theorem Proving View paper
- [59] Goedel-Prover-V2: Scaling Formal Theorem Proving with Scaffolded Data Synthesis and Self-Correction View paper
- [60] Concise: Confidence-guided compression in step-by-step efficient reasoning View paper
- [61] How to discover short, shorter, and the shortest proofs of unsatisfiability: a branch-and-bound approach for resolution proof length minimization View paper
- [62] Learning better representations from less data for propositional satisfiability View paper
- [63] Abstracting gradual typing View paper
- [64] The normalized curve shortening flow and homothetic solutions View paper
- [65] Coarsening natural deduction proofs II: finding gaunt proofs View paper
- [66] Stepwise refinement provenance scheme for wireless sensor networks View paper
- [67] Rodin: an open toolset for modelling and reasoning in Event-B View paper
- [68] Efficient, verified checking of propositional proofs View paper
- [69] Toward mechanical mathematics View paper
- [70] Draft, sketch, and prove: Guiding formal theorem provers with informal proofs View paper
- [71] Kimina-prover preview: Towards large formal reasoning models with reinforcement learning View paper
- [72] Autoformalization with large language models View paper
- [73] Formal theorem proving by rewarding llms to decompose proofs hierarchically View paper
- [74] Position: Formal Mathematical Reasoningâ□□A New Frontier in AI View paper
- [75] Towards automating formalisation of theorem statements using large language models View paper