Datasets, Formalizers, and Provers: A Survey of Formal Mathematical Reasoning with LLMs

Anonymous ACL submission

Abstract

Recent advances in large language models (LLMs) have significantly expanded the capabilities of automated mathematical reasoning. This survey reviews recent progress in formal mathematical reasoning with LLMs from three interconnected perspectives: datasets, formalizers, and provers. We provide a comparative analysis of key benchmarks from 2022 to 2025, highlighting dataset properties and model performance trends. We argue that future progress will require richer and more diverse benchmarks, evaluation protocols that emphasize semantic correctness and robustness, and deeper alignment between informal mathematical intuition and formal symbolic structure.

1 Introduction

Recent advances in large language models (LLMs) have significantly expanded the capabilities of automated mathematical reasoning. These models exhibit strong performance in solving problems stated in natural language as well as in formal logic systems. Much of the early progress has focused on informal or semi-formal problem solving, e.g., generating chain-of-thought solutions to word problems. However, recent work has shifted toward *formal mathematical reasoning*, where problem statements and proofs are expressed in formal languages and verified by proof assistants (Yang et al., 2024).

In the **informal setting**, LLMs process mathematical problems in natural language or LaTeX and output free-form solutions. This research has been accelerated by large-scale benchmarks such as GSM8K (Cobbe et al., 2021a) and MATH (Hendrycks et al., 2021a). Prompting methods including few-shot learning, chain-of-thought (CoT) prompting (Wei et al., 2022), zero-shot CoT (Kojima et al., 2022), and analogical reasoning (Zhang et al., 2023) have been introduced to elicit reasoning behavior. Numerous agent architectures—such as Xolver (Zhou et al., 2022), Toolformer (Schick et al., 2023)—combine these methods with symbolic tools or external verifiers. However, informal outputs are inherently unverifiable, making reliability and correctness difficult to guarantee.

In contrast, the **formal setting** treats reasoning as a programmatic activity within formal logic systems such as Lean (de Moura et al., 2015), Isabelle/HOL (Nipkow et al., 2002), Coq (The Coq Development Team, 1999–), or Metamath (Megill and Wheeler, 2019). Here, correctness is ensured through machine-checked proof construction, and the problem of hallucination is somewhat mitigated. In addition, feedback from prover environments can also act as feedback to the LLM so that it can refine its generation. LLMs in this domain are typically deployed in two primary roles:

- **Formalizers**, which convert informal or semi-formal mathematical statements into formal specifications. Notable systems include Kimina-Autoformalizer (Wang et al., 2025b) and RAutoFormalizer (Liu et al., 2025b).
- **Provers**, which generate or guide formal proofs, often in collaboration with interactive theorem provers (ITPs) or automated theorem provers (ATPs). State-of-the-art examples include DeepSeekProver (Xin et al., 2024; Ren et al., 2025b), KiminaProver (Wang et al., 2025b), and GödelProver (Lee et al., 2024).

The formal reasoning community has developed several benchmarks to support this research, including AIME (Analysis and Contributors, 2025), FormalMATH (Yu et al., 2025), Combibench (Liu et al., 2025a), etc. These benchmarks vary in source domain, proof assistant support, and difficulty, enabling systematic evaluation of formalization accuracy and prover competence.

A variety of modeling techniques have been applied to these tasks. Many systems leverage general-purpose LLMs (e.g., GPT-4o, DeepSeek) or domain-specific provers and formalizers, via prompting, retrieval, and stepwise inference. Beyond supervised learning, **reinforcement learning (RL)** has emerged as a key driver of progress in theorem proving. RL-based methods allow LLM agents to improve over time by interacting with proof environments and receiving feedback on tactic effectiveness, proof outcomes, and library utility—enabling adaptive search strategies and curriculum-style learning (Chen et al., 2024; Wang et al., 2024).

In this survey, we focus exclusively on formal mathematical reasoning with LLMs. Our goal is to distill and systematize recent progress across three axes:

1. **Datasets**: We analyze benchmarks and corpora that support training and evaluation of formalization and proof synthesis. In particular, we focus on the mathematical diversity in the existing datasets.
2. **Formalization Methods**: We survey techniques for translating natural language problems into machine-checkable prover-specific format.
3. **Provers and Integration Strategies**: We categorize LLM-driven proof generation methods, including tactic synthesis, guided search, retrieval-based reasoning, and reinforcement learning-enhanced exploration.

Our contributions are fourfold:

- We introduce a structured taxonomy of LLM applications in formal mathematics, distinguishing between formalization and proof generation tasks.
- We provide a comparative analysis of key benchmarks from 2022–2025, highlighting dataset properties and model performance trends.
- We synthesize methodological advances in autoformalization and LLM-prover integration, including prompting strategies, feedback loops, and reinforcement learning frameworks.
- We identify limitations in current systems—such as semantic drift, incomplete formalizations, and restricted generalization—and propose directions for scalable and verifiable formal reasoning.

2 Datasets Perspective

The evolution of large language model (LLM) approaches to mathematical reasoning can be traced through a sequence of datasets that progressively shift from informal arithmetic problem solving to fully formal theorem proving. Along this trajectory, dataset design has co-evolved with domain complexity, degrees of formalization, and evaluation paradigms—reflecting increasingly ambitious goals for machine reasoning.

Natural-language mathematical reasoning. Early benchmarks were designed to elicit step-by-step reasoning expressed in natural language. **GSM8K** (Cobbe et al., 2021b) comprises approximately 8,500 grade-school word problems requiring two to eight elementary arithmetic operations to produce a single integer answer. While mathematically simple, GSM8K exposed the brittleness of early transformer models on multi-step reasoning and played a central role in motivating chain-of-thought prompting. **MATH** (Hendrycks et al., 2021b) significantly raises the difficulty, containing 12,500 competition-style problems across pre-algebra, algebra, geometry, precalculus, number theory, and counting and probability, each annotated with a step-by-step solution and a difficulty level from 1 to 5. **OmniMATH** (Gao et al., 2024a) further extends this line by aggregating over 4,500 Olympiad-level problems across algebra, geometry, number theory, and combinatorics, curated from multiple languages and sources. By increasing both mathematical difficulty and linguistic diversity, OmniMATH challenges modern LLMs on high-level reasoning while encouraging multilingual and cross-cultural generalization. Although primarily proposed for natural-language reasoning, these datasets have also been reused as sources for formalization and proof-generation studies.

Transition to formal theorem proving. **MiniF2F** (Zheng et al., 2021) marks a decisive transition toward formal reasoning. It contains 488 problems drawn from AMC, AIME, IMO, and high-school mathematics, formalized in multiple proof assistants including Lean, Metamath, Isabelle, and HOL Light. By shifting evaluation

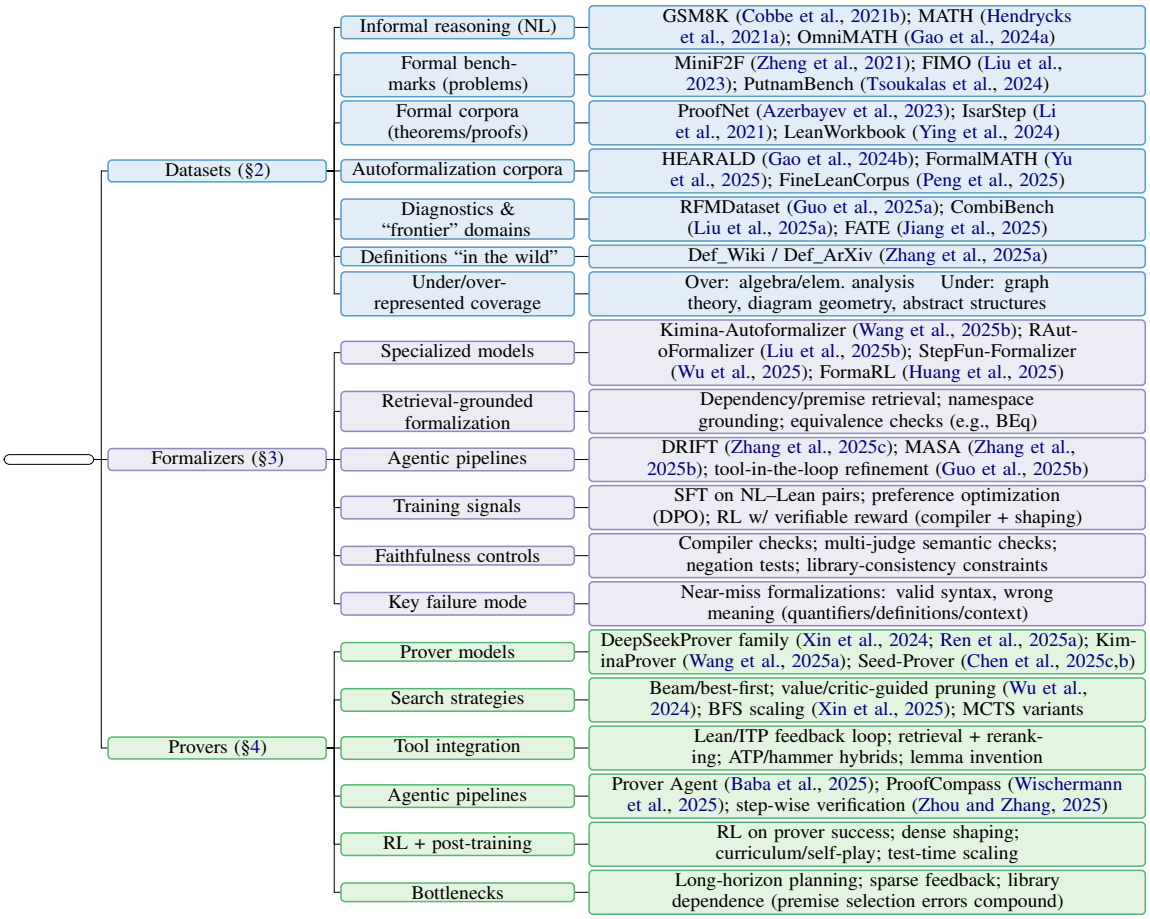

Figure 1: Revised taxonomy aligning with the survey's pipeline view: datasets enable (auto)formalization and proving; evaluation protocols (semantic fidelity, budgets, and reproducibility) determine whether progress is comparable and trustworthy.

from textual explanations to proof synthesis and formal verification, MiniF2F laid the foundation for neural theorem-proving research.

ProofNet (Azerbayev et al., 2023) expands this direction with 3,675 formally verified Lean theorems paired with human-readable proofs spanning undergraduate-level algebra, analysis, topology, geometry, number theory, combinatorics, graph theory, logic, and category theory. Carefully curated to resemble textbook-style mathematics, ProofNet exposes substantial performance gaps in neural provers: at release, best-performing models achieved only ∼2.5–3% pass@1024 on its hardest subsets. Its *ProofNet-Hard* split remains a widely used stress test.

Several datasets push formal reasoning into the most challenging domains of competition mathematics. **FIMO** (Liu et al., 2023) is a large-scale Lean 4 dataset of formalized IMO and national Olympiad problems across algebra, plane and 3D geometry, number theory, and combinatorics, particularly emphasizing invariants and extremal arguments. **PutnamBench** (Tsoukalas et al., 2024) provides formalized problems from the William Lowell Putnam Mathematical Competition, spanning analysis, algebra (including Galois theory), number theory, combinatorics, and topology. Its difficulty and origin highlight the persistent gap between informal competition statements and fully verified formal proofs.

IsarStep (Li et al., 2021) is an early dataset of 2,018 Isabelle/Isar-style proofs covering set theory, lattice theory, number theory, algebra, and analysis. By emphasizing high-level, structured proof writing rather than low-level tactic scripts, it bridges natural-language reasoning and automated proof generation. **LeanWorkbook** (Ying et al., 2024) scales this idea to over 20,000 Lean exercises extracted from undergraduate textbooks across linear algebra, real analysis, abstract algebra, topology, differential geometry, probability, graph theory, number theory, and combinatorics. Its curricu-

lar breadth makes it especially valuable for training and evaluating formal systems beyond competition-style problems.

RFMDataset (Guo et al., 2025a) takes a diagnostic perspective, curating 200 problems across combinatorics, analysis, number theory, and logic to systematically catalogue failure modes of advanced LLMs, including hallucinations, logical inconsistencies, and invalid proof steps. **HEARALD** (Gao et al., 2024b) pairs natural-language annotations with Lean 4 tactics across the full spectrum of mathlib domains, supporting autoformalization pipelines and improving prover performance via tactic alignment and human-style commentary. **FormalMATH** (Yu et al., 2025) is a large-scale Lean 4 benchmark containing 5,560 verified statements across college-level mathematics. Constructed via a human-in-the-loop autoformalization pipeline, it demonstrates that even strong models achieve only $\sim$16.5% success under practical sampling budgets, underscoring the remaining gap in formal-proof capability. **FineLeanCorpus** (Peng et al., 2025) provides nearly 286,000 verified natural-language statements paired with Lean 4 formalizations, constructed via a critic-guided human-in-the-loop pipeline. Its scale and diversity make it a central resource for autoformalization research.

CombiBench (Liu et al., 2025a) focuses exclusively on combinatorial reasoning, providing 100 Lean 4 problems from middle-school to IMO and university level, along with a fine-grained evaluation framework that moves beyond binary success metrics. **ExtremBench** (Gao and Han, 2025) is a benchmark of 93 extremal optimization problems derived from inequality exercises in Chinese Mathematical Olympiad, designed to evaluate LLMs' ability to find maxima or minima under constraints—an aspect of mathematical reasoning not captured by standard benchmarks. **Def_Wiki** and **Def_ArXiv** (Zhang et al., 2025a) shift attention from proofs to definitions, extracting formalization tasks from Wikipedia and arXiv across a wide range of mathematical fields. By emphasizing ambiguous, context-dependent formalization and library grounding, they capture a complementary aspect of mathematical reasoning.

FATE (Formal Algebra Theorem Evaluation) (Jiang et al., 2025) is a recent formal benchmark series in abstract and commutative algebra that spans from undergraduate exercises to PhD-level qualifying exam problems in Lean. The FATE series includes benchmarks such as FATE-H and FATE-X, which are explicitly designed to test models on advanced algebraic reasoning well beyond contest mathematics and standard library coverage.

3 Formalizers Perspective

Autoformalization systems, or "formalizers," represent a critical bridge between informal natural language mathematical discourse and machine-verifiable formal languages such as Lean or Isabelle. These systems can be broadly categorized into two types: **specialized formalization models**, which are fine-tuned large language models (LLMs) optimized specifically for direct translation tasks, and **formalization agents or pipelines**, which leverage modular, multi-component architectures often incorporating general-purpose LLMs, retrieval-augmented generation (RAG), theorem prover feedback loops, or multi-agent collaboration.

3.1 Formalization Models

Specialized formalization models perform end-to-end translation from informal mathematical language to prover-specific formal representations, most commonly Lean 4. These models are typically trained on curated informal–formal pairs and refined using reinforcement learning from verifiable signals such as compilation success, semantic equivalence checks, or prover feedback.

Kimina-Autoformalizer (Wang et al., 2025b) represents a strong competition-focused approach. It is initialized via supervised fine-tuning of `Qwen2.5-Coder-7B-Instruct` on curated informal–formal pairs drawn from MiniF2F, PutnamBench, ProofNet, and Compfiles, deliberately excluding Mathlib-derived data to avoid stylistic mismatch. The model is subsequently improved through expert iteration on challenging Numina-Math 1.5 problems, where candidate formalizations are filtered by Lean 4 compilation and evaluated using a high-capacity LLM judge with unanimity voting, supplemented by automated safeguards such as contradiction detection and negation testing. The resulting system achieves approximately 90% one-shot compilation and 66% judged semantic correctness on a human-curated test set, though the authors note that LLM-based evaluation remains imperfect and requires expert oversight.

RAutoFormalizer (Liu et al., 2025b) targets two persistent weaknesses in autoformalization: un-

reliable evaluation and lack of library awareness. To address the former, it introduces **BEq** (Bidirectional Extended Definitional Equivalence), a neuro-symbolic metric that assesses semantic equivalence between formal statements by grounding comparisons in formal definitions. To address the latter, RAutoFormalizer incorporates dependency-aware retrieval, constructing formalizations in topological order over library dependency graphs. The work also introduces **Con-NF**, a benchmark of 961 informal–formal pairs drawn from frontier mathematical texts to evaluate out-of-distribution generalization. Under BEq@8, RAutoFormalizer substantially outperforms prior baselines on both ProofNet and Con-NF.

StepFun-Formalizer (Wu et al., 2025) proposes a knowledge–reasoning fusion approach, motivated by the observation that autoformalization requires both syntactic fluency in the target prover language and high-level mathematical reasoning. The authors introduce the **ThinkingF** pipeline, which combines supervised fine-tuning on large-scale informal–formal pairs constructed from NuminaMath problems with reinforcement learning from verifiable rewards using reasoning-augmented traces generated by a stronger LLM. Built on DeepSeek-R1-Distill-Qwen, the resulting model achieves state-of-the-art performance on FormalMath-Lite (BEq@16 $\approx$ 60%) and ProverBench (BEq@16 $\approx$ 38%), but performs poorly on CombiBench, highlighting persistent difficulties in combinatorial domains.

FormaRL (Huang et al., 2025) explores data-efficient autoformalization via reinforcement learning on unlabeled problems. Using rewards derived from Lean compiler feedback and LLM-based semantic consistency, optimized with GRPO, FormaRL improves the pass@1 accuracy of `Qwen2.5-Coder-7B-Instruct` from 4.04% to 26.15% on ProofNet and from 2.4% to 9.6% on uproof using only 859 unlabeled problems. These results demonstrate that reinforcement learning can substantially enhance autoformalization performance even in low-data regimes.

Overall, specialized formalization models achieve strong performance on curated benchmarks through aggressive fine-tuning and reinforcement learning. However, their generalization remains limited, particularly on domains requiring deep combinatorial reasoning, rich contextual grounding, or informal conventions that diverge from benchmark-style problem statements.

3.2 Formalization Agents and Pipelines

Agentic pipelines decompose the task into iterative stages with tool use, multi-agent collaboration, or symbolic feedback and often claim to achieve superior robustness on ambiguous, graduate-level, or "in-the-wild" texts.

MASA (Zhang et al., 2025b) is a modular multi-agent framework in which specialized agents handle formalization, syntactic critique via Lean, semantic critique via an LLM, and subsequent refinement. These agents interact through critique–refinement loops augmented by library retrieval and denoising, enabling improved semantic faithfulness without task-specific fine-tuning.

DRIFT (Decompose–Retrieve–Illustrate–Formalize) (Zhang et al., 2025c) structures autoformalization as a staged pipeline that decomposes natural-language statements, retrieves relevant premises and illustrative examples, and then generates formalizations conditioned on this context, yielding notable gains on out-of-distribution problems.

ReForm (Chen et al., 2025a) introduces a reflective autoformalization framework based on Prospective Bounded Sequence Optimization, in which candidate continuations are evaluated using learned value estimates derived from prover feedback, helping reduce irreversible errors in long generation sequences.

ATLAS (Liu et al., 2025c) focuses on large-scale synthetic data generation, lifting concepts from formal libraries and applying teacher–student distillation with iterative expert revision to produce high-quality natural-language–to–Lean pairs that serve as training data for downstream autoformalizers.

FormalMATH (Yu et al., 2025) presents a human-in-the-loop autoformalization pipeline and benchmark that targets college-level and early graduate mathematics. The dataset contains over 5,500 Lean 4–verified statements spanning algebra, analysis, differential equations, probability, number theory, combinatorics, graph theory, and linear algebra. FormalMATH is constructed via an iterative pipeline in which LLMs generate candidate formal statements from informal problems, semantic consistency is checked using multiple LLM judges, and incorrect or trivial formalizations are filtered through negation testing and human review. Unlike competition-centric datasets, FormalMATH emphasizes curricular diversity and faithful statement-level formalization rather than full proof synthesis,

and serves both as a benchmark for autoformalizers and as training data for downstream prover systems.

Autoformalizer with Tool Feedback (Guo et al., 2025b) integrates live Lean compiler feedback and LLM-based semantic judging in an adaptive loop, combining synthetic pretraining with preference-based optimization to discourage unproductive revisions.

Geometry-specific pipelines (Murphy et al., 2024) incorporate diagrammatic reasoning, symbolic geometry engines, and SMT-based equivalence checking to handle implicit geometric constraints that are difficult to capture via text alone.

The **Herald pipeline** (Gao et al., 2025) constructs large-scale natural-language–annotated Lean datasets from Mathlib using retrieval-augmented, section-level translation, enabling formalization of graduate-level textbook material.

PDE-Controller (Soroco et al., 2025) applies autoformalization techniques to partial differential equations, combining decomposition, domain-specific retrieval, and symbolic verification to support applications in control and engineering.

Isa-AutoFormal (Li et al., 2024) adapts retrieval-augmented autoformalization to Isabelle/HOL by leveraging premise retrieval from Isar libraries and iterative semantic alignment. More broadly, retrieval-augmented autoformalization pipelines, such as those developed in LeanDojo (Yang et al., 2023) and related systems, demonstrate the importance of contextual grounding for reliable proof generation.

In summary, while specialized models currently dominate clean in-distribution benchmarks through aggressive fine-tuning and RL, agentic pipelines demonstrate far superior robustness on graduate-level texts and real-world mathematical documents where ambiguity, dependencies, and diagrammatic reasoning are paramount. Ongoing efforts toward unified evaluation frameworks (Mensfelt et al., 2025) aim to make future comparisons across these paradigms more reliable.

4 Provers Perspective

Formal theorem provers using large language models generate machine-verified proofs but must handle long-horizon reasoning, sparse feedback, and library-dependent search. Recent work has progressed from heuristic and sketch-based methods to reinforcement learning and agentic, retrieval-augmented pipelines.

4.1 Prover Models

Early LLM-based provers focused on generating short tactic sequences or high-level proof sketches, often relying on external automation to complete details. Systems such as **Draft–Sketch–and–Prove** (Jiang et al., 2022) demonstrated that informal reasoning steps could guide formal proof construction, but scalability to longer and more complex proofs remained limited.

Subsequent work began to incorporate learned search heuristics and reinforcement learning. **InternLM2.5-StepProver** (Wu et al., 2024) introduced a critic model trained from prover feedback to score partial proof states and guide best-first search. This marked a shift toward treating proof search as a learned decision process rather than a purely symbolic exploration.

In parallel, systems explored explicit control over search strategies. **BFS-Prover** (Xin et al., 2025) demonstrated that breadth-first exploration could be scaled effectively by combining GPU-accelerated Lean compilation with learned pruning heuristics, enabling exploration of deeper tactic trees while filtering large numbers of invalid candidates early.

More recent models emphasize large-scale synthetic data generation and reinforcement learning tightly coupled with the prover. **DeepSeek-Prover-V2** (Ren et al., 2025a) follows this paradigm by training on extensive self-generated Lean 4 proof trajectories and refining its policy through reinforcement learning from compiler feedback. Rather than explicitly separating formalization and proving, it operates directly in the formal domain and illustrates how large synthetic corpora and learned search policies can support increasingly complex proofs.

Kimina-Prover (Wang et al., 2025a) represents a complementary direction that prioritizes alignment and human-readable proofs. Trained on competition-style mathematics with explicit incentives for concise and syntactically clean proofs, it is often used as a base model for interactive or agentic proving scenarios where efficiency and interpretability are important.

Systems such as **Seed-Prover** (Chen et al., 2025c) push this trajectory further, reportedly combining large-capacity models with multi-stage reinforcement learning, lemma invention, and Lean-guided self-critique, though limited disclosure hin-

ders reproducibility. **Seed-Prover 1.5** (Chen et al., 2025b) extends this line with large-scale agentic RL and efficient test-time scaling, leveraging extensive Lean/tool feedback during training and reporting strong results on benchmarks including PutnamBench, CombiBench, and FATE.

4.2 Prover Pipelines

Beyond standalone models, an increasing body of work focuses on prover *pipelines* that orchestrate multiple components—search, retrieval, critique, and verification—around a base prover.

Some pipeline approaches integrated classical automated theorem provers with LLM guidance. **ProofCompass** (Wischermann et al., 2025) combines LLM-generated high-level proof sketches with ATP systems such as Vampire or E, using learned premise selection to bridge neural intuition and symbolic reasoning.

Agentic frameworks subsequently emerged to address hallucinations and brittle search behavior. **Prover Agent** (Baba et al., 2025) decomposes the proving process into interacting roles (e.g., planner, tactician, verifier, critic), allowing structured dialogue and critique to refine proofs incrementally under continuous Lean verification.

Several pipelines emphasize tight verification-in-the-loop. **Step-Wise Formal Verification** (Zhou and Zhang, 2025) enforces compilation and type-checking after every generated tactic, preventing error accumulation and reducing wasted exploration in long proofs.

Retrieval-augmented pipelines address the growing importance of library grounding. **REAL-Prover** (Shen et al., 2025) and related systems integrate dense premise retrieval and reranking to supply provers with relevant lemmas at each step, improving robustness on library-heavy theorems. **Aria** (Wang et al., 2025c) extends this idea by maintaining dependency graphs across entire proof projects, enabling coherent translation and verification of multi-page mathematical texts.

More recent work explores population-based and adversarial strategies. **EvolProver** (Tian et al., 2025) generates families of related theorems via symmetry and difficulty transformations to support curriculum learning and improve generalization. **GAR** (Wang et al., 2025d) introduces a generative adversarial framework in which a discriminator provides dense feedback on proof quality, encouraging more diverse exploration in sparse-reward settings.

Finally, several pipelines extend beyond pure mathematics. **Ax-Prover** (Tredici et al., 2025) applies agentic proving techniques to formal quantum physics, while **Hilbert** (Varambally et al., 2025) focuses on translating human-style informal proofs into formal Lean by hierarchically decomposing intuition into verifiable steps.

Overall, the evolution of prover systems reflects a gradual shift from single-pass generation toward tightly integrated, feedback-driven pipelines that combine learned search, retrieval, verification, and agentic control. While recent systems demonstrate impressive capabilities, challenges in long-horizon reasoning, library dependence, and reproducibility remain central to future progress.

5 Challenges and Discussion

Despite rapid progress, formal mathematical reasoning with large language models (LLMs) faces persistent challenges across the full pipeline, encompassing datasets, formalization, and automated proving. Many limitations stem not from individual components, but from their interaction: gaps in dataset coverage lead to fragile formalizations, which in turn impose excessive demands on provers. We analyze these issues from three complementary perspectives.

5.1 Dataset Coverage and Mathematical Breadth

Although recent benchmarks have grown significantly in size and ambition, they still cover only a narrow portion of mathematics. Most widely used datasets emphasize algebraic manipulation, elementary number theory, and classical analysis. Benchmarks such as GSM8K (Cobbe et al., 2021b), MATH (Hendrycks et al., 2021a), LeanWorkbook (Ying et al., 2024), and FormalMATH (Yu et al., 2025) illustrate this trend: even when multiple domains are nominally included, the majority of problems fall into a small set of familiar algebraic or analytic patterns.

The limitations of this coverage become particularly clear when considering graph theory. Several datasets claim to include combinatorics or graph-related content, yet closer inspection reveals that this coverage is shallow. For example, querying FormalMATH for graph-theoretic problems yields mostly elementary exercises, such as degree arguments or basic connectivity claims. Problems involving flows, matchings, graph minors, spectral methods, or structural decomposition—the core of

modern graph theory—are largely absent. Similarly, while CombiBench (Liu et al., 2025a) provides valuable fine-grained evaluation, it focuses on a restricted class of combinatorial tasks and does not reflect the breadth or abstraction level found in research-level graph theory.

As a result, models trained on these benchmarks risk overfitting to a narrow style of mathematics. Strong performance may indicate mastery of benchmark-specific patterns rather than genuine mathematical generality. Expanding dataset coverage—both in terms of domains and in the depth at which they are represented—is therefore essential. Without such expansion, it will remain difficult to assess whether formal reasoning systems are learning transferable mathematical principles or simply optimizing for well-worn problem templates.

5.2 Why Formalization Remains Hard

Formalizing mathematics is challenging even for human experts, and this difficulty is clearly reflected in current autoformalization systems. Informal mathematical writing often depends on implicit assumptions, shared conventions, and unstated context, all of which must be recovered and made explicit in a rigid formal language. Bridging this gap remains one of the hardest steps in the pipeline.

Most modern systems adopt a similar strategy based on iterative feedback. A model proposes a candidate formalization, receives syntactic feedback from the proof assistant and semantic feedback from equivalence checks or LLM judges, and then revises its output. This pattern underlies many recent approaches, including Kimina-Autoformalizer (Wang et al., 2025b), RAutoFormalizer (Liu et al., 2025b), DRIFT (Zhang et al., 2025c), MASA (Zhang et al., 2025b), and tool-feedback-based pipelines (Guo et al., 2025b).

Although effective on curated benchmarks, this approach has important limitations. Iterative feedback often promotes local repairs rather than genuine understanding, leading to formal statements that compile but fail to capture the intended mathematics, for instance through misplaced quantifiers or inappropriate definitions. Such near-miss formalizations are especially problematic because errors may surface only during later proof attempts, making them hard to diagnose.

More broadly, many current systems are variations on the same feedback-driven theme rather than fundamentally new paradigms. Progress beyond syntactic repair loops will likely require richer semantic representations, stronger modeling of mathematical intent, and tighter integration between informal reasoning and formal libraries, all of which remain open research challenges.

5.3 Proving Beyond Benchmarks

On the prover side, recent progress has been striking. Systems such as DeepSeekProver (Xin et al., 2024; Ren et al., 2025a), KiminaProver (Wang et al., 2025a), and Seed-Prover (Chen et al., 2025c,b) achieve strong results on benchmarks like MiniF2F and ProofNet, showing that LLM-guided proof search can successfully reconstruct many established proofs.

However, these advances are costly. Proof search is still computationally intensive, often requiring large search budgets, repeated prover interaction, and aggressive test-time scaling. More importantly, current systems rely more on extensive search than on genuine mathematical insight, compensating for limited conceptual understanding by exploring large tactic spaces guided by heuristics and dense feedback.

This raises a key open question: *how well do these methods extend to mathematics that truly requires new ideas?* Performance on research-level problems, where proofs depend on abstraction, careful lemma selection, or conceptual reformulation, remains largely untested. Closing this gap will likely require provers that operate across multiple levels of abstraction, maintain long-range coherence, and align search more closely with mathematical structure rather than surface patterns.

6 Conclusion

This survey explored formal mathematical reasoning with large language models from the perspectives of datasets, formalizers, and provers. Although recent progress driven by large-scale training, reinforcement learning, and tighter integration with proof assistants is impressive, current systems remain specialized rather than general mathematical reasoners.

We believe that advancing beyond this point will require broader and deeper benchmarks, more reliable ways to capture mathematical intent during formalization, and prover architectures that emphasize abstraction and reasoning over exhaustive search.

Limitations

This survey focuses on academic work in formal mathematical reasoning with LLMs, organized around datasets, formalizers, and provers; it omits adjacent areas and industrial systems, relies on benchmark-driven evaluations, and adopts a taxonomy that involves some subjective design choices.

Acknowledgments

We acknowledge the use of generative AI tools in the preparation of this manuscript. Specifically, generative AI tools (especially Ai2Asta and ChatGPT) were used to assist with literature search and to help polish draft text. All outputs were critically reviewed and edited by the authors to ensure accuracy and integrity.

References

Artificial Analysis and Community Contributors. 2025. AIME 2025 benchmark: Olympiad-level mathematical reasoning for LLMs. https://artificialanalysis.ai/evaluations/aime-2025. Based on the official 2025 American Invitational Mathematics Examination (AIME); 30 problems with integer answers 0–999; evaluates closed-book chain-of-thought reasoning.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stanislas Polu, and 1 others. 2023. ProofNet: Autoformalizing and formally proving undergraduate theorems. *arXiv preprint arXiv:2310.07695*.

Kaito Baba, Chaoran Liu, Shuhei Kurita, and Akiyoshi Sannai. 2025. Prover agent: An agent-based framework for formal mathematical proofs. *ArXiv*, abs/2506.19923.

Guoxin Chen, Jing Wu, Xinjie Chen, Wayne Xin Zhao, Ruihua Song, Chengxi Li, Kai Fan, Dayiheng Liu, and Minpeng Liao. 2025a. Reform: Reflective autoformalization with prospective bounded sequence optimization. *ArXiv*, abs/2510.24592.

Jiangjie Chen, Wenxiang Chen, Jiacheng Du, Jinyi Hu, Zhicheng Jiang, Allan Jie, Xiaoran Jin, Xing Jin, Chenggang Li, Wenlei Shi, Zhihong Wang, Mingxuan Wang, Chenrui Wei, Shufa Wei, Huajian Xin, Fan Yang, Weihao Gao, Zheng Yuan, Tianyang Zhan, and 3 others. 2025b. Seed-prover 1.5: Mastering undergraduate-level theorem proving via learning from experience. *Preprint*, arXiv:2512.17260.

Luoxin Chen, Jinming Gu, Liankai Huang, Wenhao Huang, Zhicheng Jiang, Allan Jie, Xiaoran Jin, Xing Jin, Chenggang Li, Kaijing Ma, Cheng Ren, Jiawei Shen, Wenlei Shi, Tong Sun, He Sun, Jiahui Wang, Siran Wang, Zhihong Wang, Chenrui Wei, and 17 others. 2025c. Seed-prover: Deep and broad reasoning for automated theorem proving. *Preprint*, arXiv:2507.23726.

Yi Chen, Zhihong Liu, Hao Li, Jing Li, Xingcheng Liu, and Maosong Sun. 2024. Realprover: A reinforcement learning-based theorem prover for large language models. *arXiv preprint arXiv:2402.12358*.

Karl Cobbe, Vineel Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021a. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021b. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. 2015. The Lean theorem prover (system description). In *Automated Deduction – CADE-25*, pages 378–388.

Binxin Gao and Jingjun Han. 2025. Max it or miss it: Benchmarking llm on solving extremal problems. *ArXiv*, abs/2510.12997.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2024a. Omni-math: A universal olympiad level mathematic benchmark for large language models. *ArXiv*, abs/2410.07985.

Guoxiong Gao, Yutong Wang, Jiedong Jiang, Qi Gao, Zihan Qin, Tianyi Xu, and Bin Dong. 2024b. Herald: A natural language annotated lean 4 dataset. *ArXiv*, abs/2410.10878.

Guoxiong Gao, Yutong Wang, Jiedong Jiang, Qi Gao, Zihan Qin, Tianyi Xu, and Bin Dong. 2025. Herald: A natural language annotated lean 4 dataset. *Preprint*, arXiv:2410.10878.

Dadi Guo, Jiayu Liu, Zhiyuan Fan, Zhitao He, Haoran Li, Yumeng Wang, and Yi R. Fung. 2025a. Mathematical proof as a litmus test: Revealing failure modes of advanced large reasoning models. *ArXiv*, abs/2506.17114.

Qianyun Guo, Jianing Wang, Jianfei Zhang, Deyang Kong, Xiangzhou Huang, Xiangyu Xi, Wei Wang, Jingang Wang, Xunliang Cai, Shikun Zhang, and Wei Ye. 2025b. Autoformalizer with tool feedback. *ArXiv*, abs/2510.06857.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Song, Michael Guo, Andy Shen, Aditya Puranik, Andy He, Andy Zou, Mantas Phan, Samuel Basart, Jacob Steinhardt, and Dawn Song. 2021a. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Yanxing Huang, Xinling Jin, Sijie Liang, Peng Li, and Yang Liu. 2025. Formarl: Enhancing autoformalization with no labeled data. *ArXiv*, abs/2508.18914.

Albert Qiaochu Jiang, S. Welleck, J. Zhou, Wenda Li, Jiacheng Liu, M. Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. 2022. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. *ArXiv*, abs/2210.12283.

Jiedong Jiang, Wanyi He, Yuefeng Wang, Guoxiong Gao, Yongle Hu, Jingting Wang, Nailing Guan, Peihao Wu, Chunbo Dai, Liang Xiao, and Bin Dong. 2025. Fate: A formal benchmark series for frontier algebra of multiple difficulty levels. *Preprint*, arXiv:2511.02872.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Sungwon Lee, Donghyun Kim, Jinwoo Shin, Junghyun Lee, Gyeongmoon Park, and Hongseok Lee. 2024. GödelProver: A neuro-symbolic theorem prover for the IMO. *arXiv preprint arXiv:2403.10101*.

Wenda Li, Lei Yu, Yuhuai Wu, and Lawrence Charles Paulson. 2021. Isarstep: a benchmark for high-level mathematical reasoning. In *International Conference on Learning Representations*.

Zenan Li, Yifan Wu, Zhaoyu Li, Xinming Wei, Fan Yang, Xian Zhang, and Xiaoxing Ma. 2024. Autoformalizing mathematical statements by symbolic equivalence and semantic consistency. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Chengwu Liu, Jianhao Shen, Huajian Xin, Zhengying Liu, Ye Yuan, Haiming Wang, Wei Ju, Chuanyang Zheng, Yichun Yin, Lin Li, Ming Zhang, and Qun Liu. 2023. Fimo: A challenge formal dataset for automated theorem proving. *ArXiv*, abs/2309.04295.

Junqi Liu, Zhibin Lin, Zhibin Zhang, Zilong Wang, Yifei Chen, Zhenyu Wang, Yu Zhang, Yu Wang, Zhiheng Zhang, Yuxiao Wang, Zihan Zhang, Ziyi Zhang, Jing Li, Kai Wang, and Zhihong Zhang. 2025a. CombiBench: Benchmarking LLM capability for combinatorial mathematics. *arXiv preprint arXiv:2505.03171*. Lean 4 benchmark with 100 combinatorial problems from middle school to IMO/university level; includes Fine-Eval framework for proof and fill-in-the-blank evaluation.

Qi Liu, Xinhao Zheng, Xudong Lu, Qinxiang Cao, and Junchi Yan. 2025b. Rethinking and improving autoformalization: Towards a faithful metric and a dependency retrieval-based approach. In *The Thirteenth International Conference on Learning Representations*.

Xiaoyang Liu, Kangjie Bao, Jiashuo Zhang, Yunqi Liu, Yu Chen, Yuntian Liu, Yang Jiao, and Tao Luo. 2025c. Atlas: Autoformalizing theorems through lifting, augmentation, and synthesis of data. *ArXiv*, abs/2502.05567.

Norman Megill and David A. Wheeler. 2019. *Metamath: A Computer Language for Mathematical Proofs*. Lulu Press.

Agnieszka Mensfelt, David Tena Cucala, Santiago Franco, Angeliki Koutsoukou-Argyraki, Vince Trencsenyi, and Kostas Stathis. 2025. Towards a common framework for autoformalization. *ArXiv*, abs/2509.09810.

Logan Murphy, Kaiyu Yang, Jialiang Sun, Zhaoyu Li, Anima Anandkumar, and Xujie Si. 2024. Autoformalizing euclidean geometry. *ArXiv*, abs/2405.17216.

Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. 2002. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer. Latest Isabelle release: Isabelle2024 (2024).

Zhongyuan Peng, Yifan Yao, Kaijing Ma, Shuyue Guo, Yizhe Li, Yichi Zhang, Chenchen Zhang, Yifan Zhang, Zhouliang Yu, Luming Li, Minghao Liu, Yihang Xia, Jiawei Shen, Yuchen Wu, Yixin Cao, Zhaoxiang Zhang, Wenhao Huang, Jiaheng Liu, and Ge Zhang. 2025. Criticlean: Critic-guided reinforcement learning for mathematical formalization. *Preprint*, arXiv:2507.06181.

Z. Ren, Zhihong Shao, Jun-Mei Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, Z. F. Wu, Zhibin Gou, Shirong Ma, Hongxuan Tang, Yuxuan Liu, Wenjun Gao, Daya Guo, and Chong Ruan. 2025a. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *ArXiv*, abs/2504.21801.

Z. Z. Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, Z. F. Wu, Zhibin Gou, Shirong Ma, Hongxuan Tang, Yuxuan Liu, Wenjun Gao, Daya Guo, and Chong Ruan. 2025b. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *Preprint*, arXiv:2504.21801.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

Ziju Shen, N. Huang, Fanyi Yang, Yutong Wang, Guoxiong Gao, Tianyi Xu, Jiedong Jiang, Wanyi He, Pu Yang, Mengzhou Sun, Haocheng Ju, Peihao Wu, Bryan Dai, and Bin Dong. 2025. Real-prover: Retrieval augmented lean prover for mathematical reasoning. *ArXiv*, abs/2505.20613.

Mauricio Soroco, Jialin Song, Mengzhou Xia, Kye Emond, Weiran Sun, and Wuyang Chen. 2025. PDE-controller: LLMs for autoformalization and reasoning of PDEs. In *Forty-second International Conference on Machine Learning*.

The Coq Development Team. 1999–. *The Coq Proof Assistant Reference Manual*. INRIA. Version 8.19 (2024), Version 8.20 expected 2025.

Yuchen Tian, Ruiyuan Huang, Xuanwu Wang, Jing Ma, Zengfeng Huang, Ziyang Luo, Hongzhan Lin, Da Zheng, and Lun Du. 2025. Evolprover: Advancing automated theorem proving by evolving formalized problems via symmetry and difficulty. *ArXiv*, abs/2510.00732.

Marco Del Tredici, Jacob McCarran, Benjamin Breen, Javier Aspuru Mijares, Weichen Winston Yin, Jacob M. Taylor, Frank Koppens, and Dirk Englund. 2025. Ax-prover: A deep reasoning agentic framework for theorem proving in mathematics and quantum physics. *ArXiv*, abs/2510.12787.

G. Tsoukalas, Jasper Lee, J. Jennings, Jimmy Xin, Michelle Ding, Michael Jennings, Amitayush Thakur, and Swarat Chaudhuri. 2024. Putnambench: Evaluating neural theorem-provers on the putnam mathematical competition. *ArXiv*, abs/2407.11214.

Sumanth Varambally, Thomas Voice, Yanchao Sun, Zhifeng Chen, Rose Yu, and Ke Ye. 2025. Hilbert: Recursively building formal proofs with informal reasoning. *ArXiv*, abs/2509.22819.

Haiming Wang, Mert Unsal, Xiaohan Lin, Mantas Baksys, Junqi Liu, Marco Dos Santos, Flood Sung, Marina Vinyes, Zhenzhe Ying, Zekai Zhu, Jianqiao Lu, Hugues de Saxc'e, Bolton Bailey, Chendong Song, Chenjun Xiao, Dehao Zhang, Ebony Zhang, Frederick Pu, Han Zhu, and 21 others. 2025a. Kimina-prover preview: Towards large formal reasoning models with reinforcement learning. *ArXiv*, abs/2504.11354.

Haiming Wang, Mert Unsal, Xiaohan Lin, Mantas Baksys, Junqi Liu, Marco Dos Santos, Flood Sung, Marina Vinyes, Zhenzhe Ying, Zekai Zhu, Jianqiao Lu, Hugues de Saxcé, Bolton Bailey, Chendong Song, Chenjun Xiao, Dehao Zhang, Ebony Zhang, Frederick Pu, Han Zhu, and 21 others. 2025b. Kimina-prover preview: Towards large formal reasoning models with reinforcement learning. *Preprint*, arXiv:2504.11354.

Hanyu Wang, Ruohan Xie, Yutong Wang, Guoxiong Gao, Xintao Yu, and Bin Dong. 2025c. Aria: An agent for retrieval and iterative auto-formalization via dependency graph. *ArXiv*, abs/2510.04520.

Hao Wang, Yuhuai Wu, Albert Q. Jiang, Xiang Li, Sean Zhu, and 1 others. 2024. Lego-Prover: Automatic theorem proving with large language models by stepwise synthesis. *arXiv preprint arXiv:2403.07890*.

Ruida Wang, Jiarui Yao, Rui Pan, Shizhe Diao, and Tong Zhang. 2025d. Gar: Generative adversarial reinforcement learning for formal theorem proving. *ArXiv*, abs/2510.11769.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Nicolas Wischermann, C. M. Verdun, Gabriel Poesia, and Francesco Noseda. 2025. Proofcompass: Enhancing specialized provers with llm guidance. *ArXiv*, abs/2507.14335.

Yutong Wu, Di Huang, Ruosi Wan, Yue Peng, Shijie Shang, Chenrui Cao, Lei Qi, Rui Zhang, Zidong Du, Jie Yan, and Xingang Hu. 2025. Stepfun-formalizer: Unlocking the autoformalization potential of llms through knowledge-reasoning fusion. *ArXiv*, abs/2508.04440.

Zijian Wu, Suozhi Huang, Zhejian Zhou, Huaiyuan Ying, Wenwei Zhang, Dahua Lin, and Kai Chen. 2024. Internlm2.5-stepprover: Advancing automated theorem proving via critic-guided search. In *unknown*.

Huajian Xin, Daya Guo, Zhihong Shao, Z. Z. Ren, Qihao Zhu, Bo Liu, Chong Ruan, Wenda Li, and Xiaodan Liang. 2024. Deepseek-prover: Advancing theorem proving in LLMs through large-scale synthetic data. *arXiv preprint arXiv:2405.14333*.

Ran Xin, Chenguang Xi, Jie Yang, Feng Chen, Hang Wu, Xia Xiao, Yifan Sun, Shen Zheng, and Kai Shen. 2025. Bfs-prover: Scalable best-first tree search for llm-based automatic theorem proving. In *Annual Meeting of the Association for Computational Linguistics*.

Kaiyu Yang, Gabriel Poesia, Jingxuan He, Wenda Li, Kristin Lauter, Swarat Chaudhuri, and D. Song. 2024. Formal mathematical reasoning: A new frontier in ai. *ArXiv*, abs/2412.16075.

Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, R. Prenger, and Anima Anandkumar. 2023. Leandojo: Theorem proving with retrieval-augmented language models. *ArXiv*, abs/2306.15626.

Huaiyuan Ying, Zijian Wu, Yihan Geng, JIayu Wang, Dahua Lin, and Kai Chen. 2024. Lean workbook: A large-scale lean problem set formalized from natural

language math problems. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhouliang Yu, Ruotian Peng, Keyi Ding, Yizhe Li, Zhongyuan Peng, Minghao Liu, Yifan Zhang, Zheng Yuan, Huajian Xin, Wenhao Huang, Yandong Wen, and Weiyang Liu. 2025. FormalMATH: Benchmarking formal mathematical reasoning of large language models. *arXiv preprint arXiv:2505.02735*. Lean 4 benchmark with 5,560 verified problems across algebra, calculus, number theory, etc.; includes autoformalization pipeline for scalability.

Lan Zhang, Marco Valentino, and Andre Freitas. 2025a. Autoformalization in the wild: Assessing LLMs on real-world mathematical definitions. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1720–1738, Suzhou, China. Association for Computational Linguistics.

Lan Zhang, Marco Valentino, and Andre Freitas. 2025b. Masa: Llm-driven multi-agent systems for autoformalization. *ArXiv*, abs/2510.08988.

Meiru Zhang, Philipp Borchert, Milan Gritta, and Gerasimos Lampouras. 2025c. Drift: Decompose, retrieve, illustrate, then formalize theorems. *ArXiv*, abs/2510.10815.

Michihiro Zhang, Ofir Press, and Yejin Wu. 2023. Analogical prompting. *arXiv preprint arXiv:2305.13574*.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. 2021. minif2f: A cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.04388*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, and Quoc V. Le. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Kuo Zhou and Lu Zhang. 2025. Step-wise formal verification for llm-based mathematical problem solving. *ArXiv*, abs/2505.20869.

Table 1: Major datasets for LLM-based mathematical reasoning (informal to formal spectrum)

Dataset	# Problems / Statements	Language / Style	Main Domains	Key Features
GSM8K	8,500	Natural language	Grade-school arithmetic word problems	First chain-of-thought benchmark
MATH	12,500	Natural language	Pre-algebra, algebra, geometry, number theory, counting	Full solutions, difficulty levels 1–5
OmniMATH	4,617	Natural language	Algebra, geometry, number theory, combinatorics	Multilingual Olympiad problems
MiniF2F	488	Lean, Metamath, Isabelle	AMC/AIME/IMO + high-school problems	First multi-prover formal benchmark
ProofNet	3,675	Lean 3/4 (human-readable)	Algebra, analysis, topology, number theory, combinatorics, logic	Clean textbook-style proofs, ProofNet-Hard subset
FIMO	843	Lean 4	Algebra, geometry, number theory, combinatorics	IMO + national Olympiad problems
PutnamBench	131	Lean 4	Analysis, algebra, number theory, combinatorics, topology	Putnam competition problems formalized in Lean
IsarStep	2,018	Isabelle/Isar	Set theory, lattices, number theory, analysis	Human-like high-level proof steps
LeanWorkbook	23,422	Lean 4	Linear algebra, analysis, topology, etc.	Undergraduate textbook exercises
RFMDataset	200	Natural language + formal	Combinatorics, analysis, number theory, logic	Failure-mode diagnostic dataset
HEARALD	1,200,000+ tactics	Lean 4 + NL commentary	All Mathlib domains	Massive tactic–comment alignment corpus
FormalMATH	5,560	Lean 4	College algebra, analysis, differential equations, probability, number theory, graph theory	Human-in-the-loop autoformalization
CombiBench	100	Lean 4	Combinatorics (counting, graph, extremal, Ramsey)	Fine-grained evaluation with Fine-Eval scoring
ExtremBench	162	Lean 4	Extremal graph / additive combinatorics, Ramsey theory, probabilistic method	Frontier extremal problems
Def_Wiki	56	Formal definitions	Wikipedia domains (groups, rings, topology, measure, etc.)	Real-world ambiguous definitions
Def_ArXiv	30	Formal definitions	arXiv papers (category theory, functional analysis, etc.)	Research-level context grounding
FineLeanCorpus	285,957	Lean 4 + NL statements	Algebra, analysis, number theory, combinatorics, topology, geometry	Largest verified autoformalization corpus (CriticLean)