AI for Mathematics: Progress, Challenges, and Prospects

Haocheng Ju[1] and Bin Dong*[2,3,4,5]

[1]School of Mathematical Sciences, Peking University, Beijing 100871, China
[2]Beijing International Center for Mathematical Research, Peking University, Beijing 100871, China
[3]Center for Machine Learning Research, Peking University, Beijing 100871, China
[4]Center for Intelligent Computing, Great Bay Institute for Advanced Study, Great Bay University, Dongguan 523000, China
[5]Zhongguancun Academy, Beijing 100094, China

May 5, 2026

Abstract

AI for Mathematics (AI4Math) has emerged as a distinct field that leverages machine learning to navigate mathematical landscapes historically intractable for early symbolic systems. While mid-20th-century symbolic approaches successfully automated formal logic, they faced severe scalability limitations due to the combinatorial explosion of the search space. The recent integration of data-driven approaches has revitalized this pursuit. In this review, we provide a systematic overview of AI4Math, highlighting its primary focus on developing AI models to support mathematical research. Crucially, we emphasize that this is not merely the application of AI to mathematical activities; it also encompasses the development of stronger AI systems where the rigorous nature of mathematics serves as a premier testbed for advancing general reasoning capabilities. We categorize existing research into two complementary directions: problem-specific modeling, involving the design of specialized architectures for distinct mathematical tasks, and general-purpose modeling, focusing on foundation models capable of broader reasoning, retrieval, and exploratory workflows. We conclude by discussing key challenges and prospects, advocating for AI systems that go beyond facilitating formal correctness to enabling the discovery of meaningful results and unified theories, recognizing that the true value of a proof lies in the insights and tools it offers to the broader mathematical landscape.

1 Introduction

The automation of mathematical reasoning has been a central objective of artificial intelligence (AI) since its inception. The dream of mechanizing reasoning predates the digital computer, tracing back to the 1920s when David Hilbert proposed a program to formalize all of mathematics, aimed at proving every theorem within a consistent axiomatic system. While this optimism was theoretically challenged by Kurt Gödel's incompleteness theorems in 1931 [66], which demonstrated that any sufficiently rich formal system is necessarily incomplete, Gödel's results did not close the subject. As noted by Ulam [144], von Neumann viewed these discoveries

*Corresponding author

First published: arXiv:2601v5 2 May 2026

as a prompt to rethink the role of formalism rather than abandon it. Later research has further shown that large portions of classical mathematics admit finitistic reductions [139], keeping the dream of partial mechanization alive.

With the advent of digital computers, these theoretical ideas were translated into practice. In the 1950s, Martin Davis implemented a decision procedure for Presburger arithmetic [34, 42], arguably the first instance of a computer mechanically verifying logical statements. Shortly after, Newell, Simon, and Shaw developed the "Logic Theorist" [119], widely considered the first true AI program, which performed theorem proving in symbolic logic. A particularly influential milestone in this symbolic era was Wen-Tsun Wu's method of mechanical theorem proving in geometry [160, 162]. By translating geometric problems into systems of algebraic equations and applying the characteristic set method, Wu demonstrated that complex logical consequences could be derived algorithmically.

However, most of these symbolic methods faced a critical bottleneck: the combinatorial explosion of the search space. As proofs grew in complexity, the number of possible logical paths expanded exponentially, rendering exhaustive search intractable. In recent decades, the integration of machine learning has revitalized this pursuit, offering data-driven approaches to navigate these complex landscapes. Since the 2010s, connectionist AI [1] has risen to prominence, achieving remarkable success in computer vision and natural language processing. Mathematicians have subsequently begun leveraging these models to identify patterns that guide human intuition [41, 71, 46], to construct counterexamples via reinforcement learning (RL) [147, 68, 137], and to train neural theorem provers [161, 13]. More recently, the rapid progress of large language models (LLMs) has enabled the generation of new mathematical constructions [130, 120, 62], autoformalization [164, 117], and collaborative theorem proving [187].

This convergence has given rise to the interdisciplinary field of AI for Mathematics (AI4Math). We emphasize that AI4Math is not merely the application of AI tools to mathematical tasks; it also encompasses the development of stronger AI systems where the rigorous nature of mathematics serves as a premier testbed for advancing general reasoning capabilities. Broadly speaking, as illustrated in Figure 1, research in this field can be categorized into two complementary directions:

- **Problem-Specific Modeling:** This involves the design of specialized architectures tailored to specific research questions or narrow classes of problems, such as guiding intuition in knot theory or reasoning within closed geometric systems. Beyond their high effectiveness for targeted tasks, these models typically require significantly less data and computational resources, making them accessible to a broader range of researchers. However, they rarely transfer to other domains without substantial modification.

- **General-Purpose Modeling:** This focuses on the development of foundation models, ranging from specialized math-specific language models to general-purpose reasoning engines, designed to support broader workflows across diverse mathematical areas. While offering versatility, these approaches demand massive training datasets, substantial computational resources, and significant engineering expertise. Furthermore, they may not achieve the same degree of specificity or effectiveness as specialized models when applied to narrowly defined mathematical problems. This category encompasses advances in natural language reasoning, the bridging of informal and formal mathematics through autoformalization, and the construction of agentic systems capable of automated theorem proving and information retrieval.

It is worth noting that the scope of AI4Math technically extends beyond logical reasoning to include AI for Computational Mathematics and Scientific Computing. This domain focuses

[1] Connectionist AI is an AI paradigm inspired by the interconnected structure of neurons in the human brain. It enables perception, reasoning, and decision-making through multi-layer networks of nodes that learn patterns from data by optimizing connection weights.

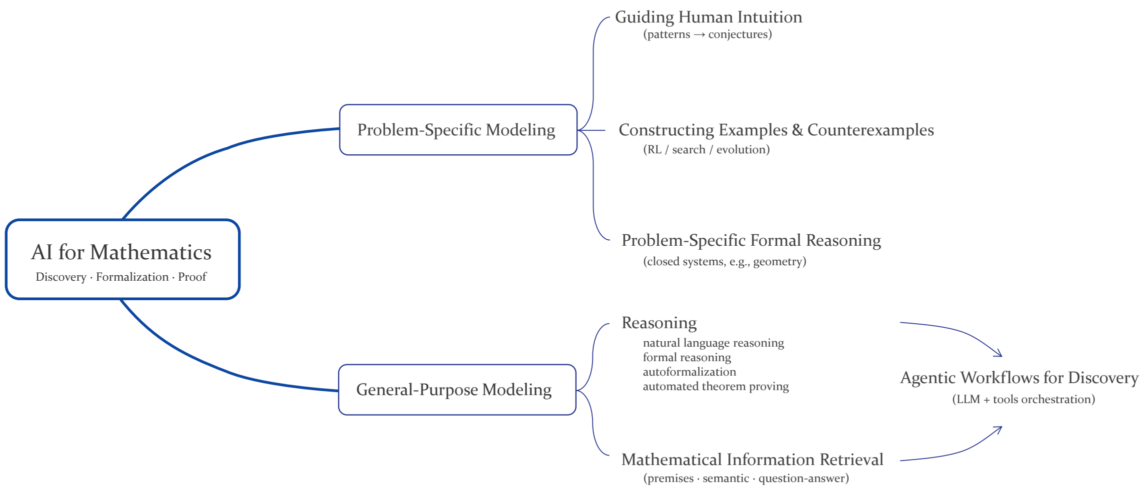

Figure 1: Landscape of AI4Math research directions. AI4Math research can be broadly divided into two complementary paradigms: *problem-specific modeling* and *general-purpose modeling*. Problem-specific modeling encompasses three major lines of work: using machine learning to guide human intuition, constructing examples and counterexamples, and performing formal reasoning in closed systems. General-purpose modeling focuses on mathematical reasoning and mathematical information retrieval; together, these capabilities underpin emerging agentic workflows for mathematical discovery. This diagram is intended to organize and clarify conceptual relationships among research directions, rather than to provide a one-to-one correspondence with the sections of this review.

on building AI models to assist in numerical computations, such as for solving partial differential equations (PDEs), optimization, and inverse problems. The roots of this approach trace back to the 1980s and 1990s, where researchers explored the use of shallow neural networks to approximate solutions to differential equations. However, the field experienced a renaissance around 2015 with the advent of deep learning. While these computational advances constitute a vital pillar of the broader AI4Math landscape, this review restricts its focus to mathematical reasoning—encompassing discovery, formalization, and proof. We refer readers interested in the computational aspects of AI4Math to other comprehensive surveys [8, 88, 21, 159, 45, 28].

In this paper, we provide a systematic overview of the progress, challenges, and prospects of AI4Math. Our goal is not to be exhaustive, but to highlight representative works that illustrate the field's evolution. Readers interested in specific subareas may consult existing surveys on automated reasoning in geometry [32, 163] or deep learning for theorem proving [105]. Section 2 examines problem-specific modeling, while Section 3 reviews general-purpose modeling. Finally, Section 4 discusses the key challenges ahead, advocating for systems that move beyond simple verification toward the discovery of deep mathematical insights.

2 Problem-Specific Modeling

With the rapid advancement of data-driven techniques, researchers started to design specialized machine learning models tailored to specific mathematical research problems. These efforts generally fall into three distinct directions: identifying patterns in high-dimensional data to guide human intuition and inspire new conjectures; constructing examples or counterexamples to rigorously test or disprove mathematical hypotheses; and performing formal reasoning within closed axiomatic systems, such as Euclidean geometry. In this section, we review recent developments across these three areas and discuss their respective advantages and limitations.

2.1 Guiding Human Intuition through Machine Learning

One of the earliest examples of utilizing machine learning to assist in formulating mathematical conjectures is [25]. In this study, the authors employed linear regression to predict the rank of the geometric gauge group across a large ensemble of F-theory geometries, successfully rediscovering an existing conjecture regarding gauge group rank. Building on this, they applied logistic regression to a classification problem involving E_6 gauge groups; through attribution analysis of the model, they formulated a genuinely new conjecture that was subsequently proved by human mathematicians.

However, the pivotal study that demonstrated the broad potential of deep learning in mathematical research is [41]. The central contribution of this work is a systematic framework designed to accelerate the conjecture generation process—traditionally a time-consuming cycle where mathematicians hypothesize relationships, attempt proofs, and refine their ideas iteratively. In the workflow proposed by [41], mathematicians begin by hypothesizing that a relationship may exist between two mathematical objects. A specially designed neural network is then trained to predict one quantity from the features of the other. Attribution methods are subsequently used to identify the most influential input components, thereby guiding mathematicians toward more precise and refined conjectures. This cycle repeats until a mathematically meaningful statement is formulated. Using this approach, the authors discovered a new relationship between algebraic and geometric invariants in knot theory [40] and proposed a candidate algorithm predicted by the combinatorial invariance conjecture for symmetric groups [19].

Inspired by this paradigm, [46] designed another AI-guided intuition framework to the study of *affine Deligne–Lusztig varieties (ADLV)*. This work not only independently rediscovered the classical virtual dimension formula in arithmetic geometry but also established a novel, precise lower-bound theorem. By providing a quantitative characterization that fills an important theoretical gap, this result demonstrates the efficacy of *AI-guided intuition* in facilitating rigorous discoveries within deep areas of pure mathematics.

Beyond refining conjectures, machine learning has also proven effective in uncovering entirely new mathematical phenomena. A notable example is [72], where the authors represented elliptic curves as vectors and trained a logistic regression classifier to distinguish curves of different ranks. To interpret the classifier's strong performance, the authors performed Principal Component Analysis (PCA) on the vector representations. Upon plotting the averages of Frobenius traces for elliptic curves of fixed rank over a given conductor range, they revealed a surprising oscillatory pattern, which they termed "murmurations." This phenomenon has since been the subject of in-depth theoretical study [189, 101, 20]. A growing body of literature continues to leverage machine learning to discover or rediscover relationships across various domains [25, 90, 51, 16, 18, 99, 7, 69, 134, 96, 100].

2.2 Constructing Examples and Counterexamples

One of the pioneering works to utilize machine learning for constructing examples and counterexamples is [147]. In this study, the author encodes graphs as 0–1 sequences and applies the deep cross-entropy method from RL to search for graph constructions that serve as counterexamples to existing conjectures. Following this precedent, RL has been adopted for a variety of structural problems. For instance, [15] models the Hironaka game, a framework central to the resolution of singularities in algebraic geometry, as a Markov decision process (MDP). By combining Monte Carlo tree search with Deep Q-Networks, the authors successfully trained an agent to replace singular points with smooth ones, achieving near-optimal resolutions for du Val singularities. Similarly, [137] investigates the *Andrews–Curtis (AC)* conjecture. After first employing classical search algorithms to identify AC triviality for infinite subfamilies of the *Miller–Schupp* series and to obtain length reductions in the *Akbulut–Kirby* series, the authors formulate the general problem as an MDP. They trained an RL agent on problem instances of varying difficulty, ultimately discovering two novel AC trivializations of balanced presentations that had eluded classical search methods.

Beyond RL, a diverse array of other machine learning techniques has been proposed for mathematical construction. [6] trains a Transformer model on synthetic data to predict Lyapunov functions for stable dynamical systems; the trained model is then used to discover novel Lyapunov functions for non-polynomial systems. In combinatorics, [26] introduces an iterative *bootstrapping* procedure. This method begins by generating candidate constructions via a local search algorithm, trains a neural network on the highest-scoring candidates, and then samples new seeds from the network to initialize the next round of local search. Using this approach, the authors successfully discovered a counterexample to a 30-year-old conjecture. [23] proposes a geometry-aware conditional flow-matching model for continuous geometric optimization problems, interleaving flow integration with projection onto the constraint manifold. The method initializes a training set using stochastic relaxation with perturbations, trains a conditional flow-matching model with geometric penalties, and employs geometry-aware sampling together with reward-guided policy optimization to iteratively refine configurations via local search. This approach discovers configurations that match or exceed the best known results in problems such as circle packing maximizing the sum of radii and the Heilbronn triangle problem. Additionally, [17] applies genetic algorithms to generate reflexive polytopes in dimensions 2 through 5, identifying several previously unknown polytopes in dimension 5. Other works leveraging machine learning for the construction of examples or counterexamples include [30, 31, 68, 140, 48].

As LLMs continue to grow more capable, LLM-based agents have also improved, and several recent works have demonstrated their potential for discovering new mathematical constructions. *FunSearch* [130] searches for programs that generate desired constructions using an evolutionary approach. For problems equipped with a well-defined evaluator, FunSearch employs an off-the-shelf LLM to iteratively evolve low-scoring candidate programs into higher-scoring ones. This is achieved by maintaining a large, diverse pool of programs and repeatedly prompting the LLM to improve upon earlier candidates. Using this method, FunSearch discovered new constructions of large cap sets that surpass the best previously known results in extremal combinatorics. Building on this line of work, *AlphaEvolve* [120] uses a stronger LLM and extends the evolutionary process to entire code files rather than single functions, while also optimizing multiple metrics simultaneously. AlphaEvolve has produced improved constructions for several problems, including the *Minimum Overlap Problem* and the *Kissing Numbers problem* in 11 dimensions. Open-source implementations inspired by AlphaEvolve include *OpenEvolve* [136], *ShinkaEvolve* [97], and *DeepEvolve* [107]. AlphaEvolve-style agents are particularly suitable for finding new constructions in mathematical problems that can be approached by writing code and evaluated with a well-defined scoring function.

2.3 Problem-Specific Formal Reasoning

AlphaGeometry [143] represents a neuro-symbolic approach to solving Olympiad-level Euclidean geometry problems. It integrates a symbolic geometric reasoning engine with a language model designed to suggest auxiliary constructions. The symbolic component, based on a deductive database (DD) [33] and algebraic rules (AR), exhaustively derives the deduction closure of a given set of premises. Since purely symbolic deduction cannot introduce new geometric objects, a capability often required for complex proofs, the system utilizes a language model to propose useful auxiliary points. To circumvent the scarcity of human proof data, the authors generated a large-scale dataset of synthetic proof graphs by alternating symbolic deduction with random point insertions, subsequently extracting minimal proofs via traceback. During inference, the system operates in a loop: the symbolic engine expands the deduction closure, the model proposes high-probability auxiliary points, and the cycle repeats until the target conclusion is reached. This architecture allows AlphaGeometry to significantly outperform heuristic-based systems, reaching performance comparable to an IMO silver medalist.

Its successor, *AlphaGeometry2* [29], further advances this paradigm through enhancements in both expressivity and efficiency. The formal geometry language was expanded to support locus descriptions, linear geometric relations, and non-constructive statements, while the underlying symbolic engine was re-engineered for greater speed and robustness. These improvements enabled the generation of a larger, more diverse synthetic training set for the language model. Furthermore, AlphaGeometry2 introduces a novel search algorithm, the *Shared Knowledge Ensemble of Search Trees (SKEST)*, which executes multiple search trees in parallel. By allowing these trees to exchange discovered information, SKEST significantly improves the exploration of the auxiliary construction space. Consequently, AlphaGeometry2 achieves gold-medalist performance on IMO-level geometry problems. Beyond neural approaches for auxiliary point generation, recent work such as *HAGeo* [50] proposes a purely heuristic-based strategy that introduces auxiliary constructions with favorable geometric properties, including intersections of lines and circles, midpoints, and point reflections, and likewise achieves gold-medal-level performance. Additional work on Euclidean geometry problem solving can be found in [181, 73, 188, 182, 177].

2.4 Discussion

The approaches discussed in this section, identifying patterns to guide intuition, constructing counterexamples, and performing formal reasoning in closed systems, each offer distinct advantages while presenting unique challenges.

The *AI-guided intuition* paradigm is powerful because it enables mathematicians to uncover patterns in high-dimensional data that are otherwise difficult or time-consuming to detect manually, effectively reducing the search space in exploratory research. However, this approach is not universally applicable. It relies on careful problem selection, as the target question must admit the generation of sufficiently large and representative datasets. Furthermore, successful implementation demands a high barrier of dual expertise: beyond standard machine learning considerations such as architecture design and loss engineering, deep mathematical insight is indispensable for interpreting model outputs and transforming empirical correlations into rigorous mathematical theory. Ultimately, since the final proof and verification are typically carried out by human mathematicians, the degree of automation in this workflow remains limited.

Conversely, employing machine learning to construct examples and counterexamples can significantly accelerate the formulation and testing of conjectures by discovering objects that defy human intuition. Yet, this direction also faces technical hurdles, particularly regarding out-of-distribution (OOD) generalization. For instance, the backward generation method in [6], which constructs problems from sampled solutions, may yield training sets with specific distributions. This can present challenges for generalizing to typical problem instances and often requires carefully designed procedures, such as the diversity promoting mechanism for Lyapunov functions proposed by the authors, to ensure robust performance. Additionally, when RL is employed, mapping a mathematical problem onto an MDP is non-trivial. Defining appropriate state representations, action spaces, and reward functions can be intricate [147], and the learning process may be further complicated by sparse rewards and long planning horizons [137].

Systems like FunSearch and AlphaEvolve represent a powerful shift toward automated mathematical discovery through program evolution. These agents excel at construction-type problems, such as finding extremal configurations in combinatorics or improving algorithmic bounds, where the search space can be navigated by iteratively refining code and verified through a well-defined scoring function. However, this paradigm remains inherently constrained by its reliance on such evaluators. While highly effective at identifying specific mathematical objects (e.g., cap sets or high-dimensional sphere packings), these systems do not explicitly engage with the formulation of conceptual frameworks or structural principles that underlie general theories or connect disparate mathematical domains. Their success is therefore currently concentrated in verifiable construction, leaving the discovery of transferable abstractions and unifying insights as an open frontier.

Finally, problem-specific formal reasoning systems, such as AlphaGeometry, demonstrate that combining symbolic engines with neural language models can achieve expert-level performance in structured domains. However, the success of these systems typically relies on the availability of domain-specific symbolic engines (e.g., deductive databases for geometry [33]) and the capability to generate large-scale synthetic data. Consequently, these architectures are often tightly tailored to their specific problem scope and may not transfer to other areas of mathematics without substantial modification.

3 General-Purpose Modeling

General-purpose modeling marks a shift from specialized algorithms designed for isolated problems to adaptable systems capable of handling broad areas of mathematics. Unlike problem-specific modeling, which requires bespoke features and architectures for each new task, general-purpose approaches leverage foundation models trained on vast corpora to learn universal representations of mathematical knowledge. These models aim to support a wide spectrum of activities, from solving diverse problem sets to retrieving theorems and orchestrating complex workflows, without requiring substantial modification for each new domain.

We categorize recent progress in general-purpose modeling into three complementary directions: natural language reasoning models that leverage the intuitive power of language; formal reasoning models that ensure rigor through interaction with proof assistants; and mathematical information retrieval systems that ground reasoning in established knowledge. This section begins by analyzing the capabilities and inherent limitations of foundation models (especially LLMs), establishing the context for our detailed review of these three key areas.

3.1 Foundation Models and LLMs

Compared with traditional machine learning models that are trained for a single, narrowly defined task, LLMs serve as foundation models: a single architecture trained on a broad collection of data and tasks in a unified manner. Mathematically, this distinction represents a paradigm shift from function approximation to operator approximation, which is a process intimately related to *meta-learning*. A traditional model is typically designed to approximate a specific, fixed mapping $f : \mathcal{X} \to \mathcal{Y}$ between Euclidean spaces, such as classifying objects in an image, by interpolating discrete data points sampled from that specific manifold.

In contrast, foundation models perform *task interpolation* rather than simple data interpolation. They are trained over a vast distribution of tasks $\{T_i\}$, where each task effectively defines a distinct mapping f_i. By learning to generalize across this distribution, the foundation model does not merely approximate a single function; instead, it approximates a global operator (or functional) Ψ acting on a function space. Given a task specification or context as input, Ψ instantiates the specific mapping f_i required for that context. In essence, while traditional machine learning approximates functions, foundation models approximate the operators that generate and manipulate those functions.

A critical factor in their success is the capability to process diverse data types; tokenization converts heterogeneous inputs into a common sequence representation, and the next-token prediction objective provides a single learning rule that applies uniformly across all tasks the model encounters. Attention-based architectures are pivotal in this regime. Beyond effectively scaling with model size and data volume, the attention mechanism serves as the key engine for reasoning by enforcing long-context coherence during training. This allows the model to capture and maintain complex dependencies over extended sequences—a prerequisite for logical deduction. Through exposure to diverse domains and supervision signals, the model is forced to compress vast amounts of heterogeneous data into shared internal representations and to discover a common low-dimensional structure across tasks and languages. A natural hypothesis is that a key component of this low-dimensional structure corresponds to general-purpose reasoning abilities that can be expressed in different languages and domains.

Mathematics fits naturally into this framework. Mathematical work is governed by strict logical rules, and many mathematical tasks can be phrased as producing the next meaningful step in a calculation, derivation, or proof—exactly the kind of stepwise structure that the next-token prediction objective is designed to model. Consequently, when an LLM is trained as a foundation model on sufficiently rich mathematical and scientific corpora, the same mechanisms that support cross-domain generalization and long-context coherence can be harnessed to learn and use a wide range of mathematical reasoning abilities.

However, a fundamental gap remains between mastering standardized examinations and engaging in research-level mathematics. While current models excel at solving well-defined undergraduate problems, research demands open-ended exploration, absolute logical rigor, and the ability to navigate the "long tail" of specialized domain knowledge—challenges where stochastic text generation often falls short. To elevate AI from a competent solver to a reliable research partner, general-purpose modeling must therefore advance beyond simple next-token prediction, integrating formal verification, semantic retrieval, and agentic workflows to bridge the divide between plausible text and rigorous truth.

3.2 Natural Language Reasoning

Current approaches to natural language mathematical reasoning generally fall into two categories: *math-specific LLMs* and *general-purpose reasoning models*.

Math-specific LLMs are typically adapted from general foundation models through specialized pre-training and post-training pipelines. During pre-training, filtering pipelines [135, 171] extract high-quality mathematical content from web corpora (e.g., Common Crawl), textbooks, and research papers to maximize domain relevance. Post-training refines these models using supervised fine-tuning (SFT) and RL. The data for SFT is often structured as Chain-of-Thought (CoT) [158] pairs, consisting of problems and step-by-step solutions, or Tool-Integrated Reasoning (TIR) [67] examples that incorporate external code execution. A prominent example is *NuminaMath* [103], which won the first AIMO Progress Prize by fine-tuning on high-quality CoT and TIR datasets. While models in this category [12, 135, 171, 174] excel at elementary and competition-level benchmarks (e.g., GSM8K [35], MATH [74], AIME), their capacity for advanced mathematics remains less explored.

Concurrently, general-purpose LLMs have achieved significant mathematical milestones driven by scale and novel inference strategies. Early iterations like GPT-3 struggled with basic arithmetic, whereas GPT-4 [2] achieved 92.0% on GSM8K. A paradigm shift occurred with the introduction of *test-time scaling*, where models dedicate increased computation to reasoning during inference. OpenAI's o1 model demonstrated strong performance on the AIME, and subsequent reasoning models [70, 142, 170, 37] have further validated this approach. By 2025, enhanced reasoning models, such as Google's Gemini Deep Think, reached gold-medal performance at the International Mathematical Olympiad (IMO) using purely natural-language reasoning, marking a definitive maturity for the technology in the domain of high-school olympiad mathematics.

However, transitioning from Olympiad problems to higher mathematics presents a steeper challenge. Prior studies indicate that while GPT-4 can assist with undergraduate topics, it requires critical human oversight [36] and often fails at the graduate level [57]. Recent benchmarks quantify this gap: [85] reported that DeepSeek-R1 achieves 71.0% proof accuracy on graduate-level algebra (FATE-H) but drops to 33.0% on PhD qualifying exams (FATE-X). Similarly, on the *FrontierMath* Benchmark [63], which consists of unpublished research-level problems, Gemini 3 Pro scores 18.75% on the research-level split (Tier 4)[2], indicating that robust research-level

[2] https://epoch.ai/frontiermath

reasoning remains an open problem.

To empirically assess the current capabilities of state-of-the-art models, we constructed a dataset of 100 problems drawn from the final exams of 11 undergraduate mathematics subjects at Peking University (PKU). We evaluated five models: GPT-4, o1, o3-mini, DeepSeek-R1, and Gemini 2.5 Pro. Sample problems and model responses are provided in Appendix B. Human experts graded the outputs on a 0–5 scale (criteria in Table A); the normalized results are presented in Figure 2 (left and middle). While GPT-4 scores below 60, reasoning-enhanced models (OpenAI o-series, DeepSeek-R1, Gemini 2.5 Pro) show substantial improvement, with several exceeding a score of 90.

Furthermore, we evaluated o3-mini on 58 problems from PKU PhD Qualifying Exams in Analysis, Probability, Algebra, and Geometry & Topology. As shown in Figure 2 (right), o3-mini achieves an average score of 84.4. A closer examination of the subject-wise performance reveals a notable divergence: the model exhibits its strongest proficiency in Algebra while scoring lowest in Geometry & Topology. Under the assumption that these exams pose a comparable level of challenge for human students, this suggests that current AI systems are comparatively more adept at handling abstract algebraic structures than tasks requiring geometric intuition. Although these results must be interpreted with caution due to potential data contamination and the specific nature of exam problems compared to open research, they provide strong evidence that top-tier models can now handle a significant portion of graduate-level mathematics.

Building on frontier models, LLM-based agents equipped with appropriate harness engineering can further extend performance toward research-level mathematics. One representative example is *Aletheia* [54], an agent built on an advanced version of Gemini Deep Think that coordinates separate generator, verifier, and reviser components. It has been applied to several research-level tasks, including the computation of eigenweights for the Arithmetic Hirzebruch Proportionality Principle [52], questions about the simplicity of the Hodge bundle [122], and bounds for independence sets [98]. In addition, *Aletheia* solved 6 of the 10 problems in the First-Proof benchmark [53], introduced in [1], a collection of research problems that had been solved by human mathematicians but were not publicly available at the time of evaluation. Another example is *Rethlas* [87], an agent designed to mimic the workflow of human mathematicians by combining exploratory reasoning with tools such as theorem retrieval. This framework automatically resolved an open problem in commutative algebra proposed by D. D. Anderson in 2014 [24].

Synthesizing these findings, we observe a clear trajectory: the mathematical reasoning abilities of LLMs and LLM-based agents have progressed from mastering elementary calculations and high-school competitions to achieving competence in undergraduate curriculum, and are now beginning to penetrate the domain of graduate and research-level mathematics [22, 81].

3.3 Formal Reasoning

Although top-tier LLMs and LLM-based agents can now solve certain graduate-level and even certain research-level problems, evaluating their capabilities remains a significant bottleneck involving substantial human effort. As the complexity of mathematics increases, assessment depends heavily on domain experts; yet, because natural language is inherently imprecise, even experienced mathematicians can misjudge arguments. A notable historical example occurred in 1994, when a result published in the *Annals of Mathematics* claimed that the Busemann–Petty problem has a negative solution in dimensions at least four [178]. This conclusion was later shown to be incorrect [92, 91], and a 1999 paper established that the problem in fact possesses a positive solution in dimension four [179]. Similarly, a paper [132] published in the *Annals of Mathematics* in 2004 was retracted in 2023 [133] after a counterexample to its main result, discovered in 2006, was identified [93]. These cases illustrate that errors can persist even within the rigorous peer-review processes of top journals. To enable fast and reliable verification of mathematical reasoning, research must therefore move toward a more mechanical and unam-

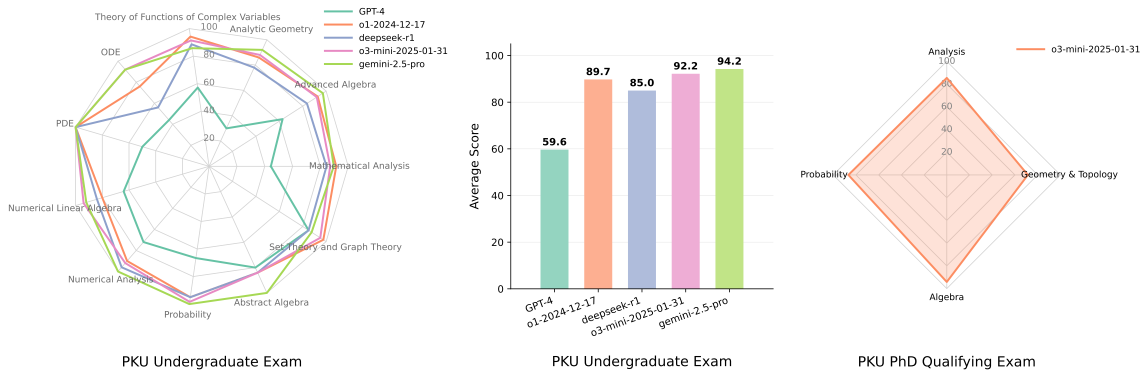

Figure 2: Performance of five LLMs across different Peking University (PKU) exams. Left: Scores on 11 undergraduate subjects. Middle: Average scores of the five LLMs across these subjects. Right: o3-mini's performance on four PhD qualifying exam fields (average score: 84.4).

biguous framework. Formal systems provide exactly such a foundation. In this section, we discuss the benefits of formal systems for mathematical research and their utility in enhancing the reasoning abilities of LLMs, followed by a review of recent progress in automated theorem proving and autoformalization.

3.3.1 Formal Systems

Formal systems provide a precise symbolic language paired with rigorously defined mechanisms for constructing and verifying proofs. Various systems exist, distinguished by their underlying logical foundations: HOL systems (e.g., HOL Light, Isabelle/HOL) utilize simple type theory; Coq (now known as Rocq) and Lean employ dependent type theory; Metamath operates on first-order logic with explicitly specified axioms; and Mizar is grounded in Tarski–Grothendieck set theory. Once a mathematical argument is translated into the formal language of an interactive theorem prover (ITP), it can be verified with absolute rigor. Had the erroneous 1994 result regarding the Busemann–Petty problem been formalized in such a system, the underlying logical flaws would likely have been detected immediately, preventing the publication of an incorrect claim.

Beyond their intrinsic value to mathematical correctness, formal systems offer a critical advantage for AI development: they provide reliable, verifiable supervision. Unlike elementary mathematics, where answers are often numeric and easily checked, advanced proof-oriented problems lack simple verifiers, making it difficult to generate reliable training signals. Interactive theorem provers bridge this gap by providing precise feedback on the validity of every logical step. This capability supplies high-quality training signals for RL, thereby enabling models to develop stronger reasoning capabilities in rigorous settings.

Among interactive theorem provers, Lean [44, 43] has cultivated a particularly robust ecosystem. Its unified mathematical library, *mathlib4*, has expanded rapidly through large-scale community collaboration and, as of December 2025, contains over 250,000 theorems and 120,000 definitions. A landmark achievement in this domain is the *Liquid Tensor Experiment*, led by Johan Commelin, which formalized a central theorem of Peter Scholze on liquid vector spaces. Scholze, who initially harbored doubts about the proof's correctness, remarked that the theorem might be his "most important to date"[3]. The project, which spanned approximately 18 months, not only verified the result but also led to a simplification of the original Clausen–Scholze proof, helping Scholze deepen his understanding of the argument's structure[4]. Furthermore, the Liquid

[3] https://xenaproject.wordpress.com/2020/12/05/liquid-tensor-experiment/
[4] Xena Project blog post, June 2021

Tensor Experiment catalyzed the development of algebraic infrastructure within mathlib4: it drove the early formalization of homological algebra and category theory and attracted a wave of algebraists to the community. Other notable milestones include the formalization of the Polynomial Freiman–Ruzsa (PFR) conjecture[5], the sphere eversion theorem[6], ongoing work on formalizing the solution to the sphere packing problem in dimension 8[7], and the long-term effort to formalize Fermat's Last Theorem[8] lead by Kevin Buzzard. Expanding into applied mathematics, recent work has also established the formal foundations for numerical optimization in Lean4, specifically verifying the convergence of first-order algorithms [102].

However, these projects remain labor-intensive, requiring experts to manually translate definitions and proofs into code. This high cost motivates the development of **autoformalization** tools and **automated theorem provers** to accelerate the *digitalization* of mathematical knowledge—the conversion of standard informal mathematics into rigorous formal systems such as Lean.

3.3.2 Autoformalization

Autoformalization is the task of translating mathematical statements and proofs from natural language into formal code in an autonomous manner (e.g. by a language model). Early efforts in this domain employed sequence-to-sequence models trained on aligned data. For instance, [155] constructed datasets by informalizing Mizar statements to train translation models. Addressing the scarcity of aligned pairs, [154] subsequently explored unsupervised approaches based on cycle-consistency losses, where models learned to reconstruct statements by translating them between informal and formal domains without explicit supervision.

The advent of LLMs fundamentally shifted this paradigm. Research demonstrated that off-the-shelf LLMs could generate reasonable formalizations via few-shot prompting [164, 58, 4, 11]. Crucially, [164] observed an asymmetry: translating formal code to natural language (informalization) is significantly easier for models than the reverse. This insight catalyzed the creation of large-scale synthetic datasets, where LLMs are used to informalize massive formal libraries (such as mathlib4) to generate high-quality aligned pairs for training specialized autoformalizers [82, 112, 60, 109].

Recent work focuses on improving the quality and grounding of these systems. *Herald* [60] introduces a hierarchical informalization strategy that respects the dependency graph of the library mathlib. By translating declarations in topological order, it ensures that the natural language descriptions of prerequisite concepts are available when translating dependent theorems. Herald further augments its data via tactic-based state synthesis and achieves over 96% statement autoformalization accuracy on the miniF2F validation set. To enhance grounding, *RAutoformalizer* [109] employs retrieval mechanisms to anchor generated code in existing formal declarations. Addressing the challenge of missing concepts, common in research-level mathematics, [151] proposed *Aria*, an LLM-based agentic system. More generally, an LLM-based agent refers to a system that operates through an explicit interaction loop with its environment: it maintains intermediate states, reasons over observations, performs multi-step planning, and selects actions accordingly. These actions may include invoking external tools such as semantic retrieval, symbolic reasoning modules, or code synthesis components, with feedback from the environment used to guide subsequent decisions. Such agentic designs enable the decomposition of complex tasks into structured subtasks and support iterative refinement beyond single-pass generation [152]. Within this framework, Aria decomposes informal statements into a dependency graph of concepts; if a concept is absent from the library mathlib, the agent recursively

[5]https://teorth.github.io/pfr//
[6]https://leanprover-community.github.io/sphere-eversion/
[7]https://thefundamentaltheor3m.github.io/Sphere-Packing-Lean/
[8]https://leanprover-community.github.io/blog/posts/FLT-announcement/

defines it from the bottom up using semantic search and synthesis, effectively handling the "long tail" of mathematical vocabulary.

Evaluation and Verification Evaluating statement autoformalization is non-trivial. Once a statement is correctly autoformalized, evaluating an autoformalized proof becomes straightforward due to the formal system, which makes statement autoformalization the primary difficulty from an evaluation perspective. While human expert review is the gold standard, it is unscalable. The challenge, therefore, lies in developing automated metrics for correctness.

- **With Ground Truth:** When a reference formal statement exists, correctness can be assessed via logical equivalence rather than mere string matching. For example, *BEq* [109] utilizes a neural theorem prover to check if the generated statement and the ground truth can be derived from one another. Similar equivalence-checking approaches are explored in [110, 124].
- **Without Ground Truth (Semantic Verification):** In the absence of a reference, one must verify *semantic correctness*—whether the formal code faithfully captures the intent of the informal statement. A naive approach involves "back-translation", where an LLM translates the code back to English for comparison [173, 60]. However, this is prone to errors as LLMs may miss subtle logical discrepancies. To mitigate this, [169] proposes *Mathesis*, a fine-grained evaluation framework. Mathesis decomposes statements into assumptions and conclusions, evaluates the alignment of each component separately, and aggregates these scores using a fuzzy integral to rigorously reject inconsistencies. To further assist verification, Aria [151] enriches the context by retrieving detailed metadata (types, values, informal descriptions) for every term in the formal statement, enabling more accurate semantic judgments.

Reliable verifiers are not only essential for evaluation but also serve as critical reward models for RL, creating a feedback loop that iteratively improves autoformalization performance [169, 112, 77].

Note: This section has focused on *statement* autoformalization. The autoformalization of *proofs*, which requires translating logical reasoning steps rather than just definitions, is inextricably linked to automated theorem proving. We therefore discuss proof autoformalization in the context of proof generation in the following section.

3.3.3 Automated Theorem Proving

Automated theorem proving in formal systems aims to generate valid proofs for formal statements. Deep learning–based approaches to this task fall into two broad categories: *single-model approaches* and *agentic approaches.* Single-model approaches can be further divided into *proof step generation* and *whole-proof generation* methods.

Proof Step Generation Proof step generation methods formulate theorem proving as a tree search problem. In this framework, each node in the search tree corresponds to a proof state, and each action corresponds to the application of a tactic that transitions the prover to a new proof state. A proof is successfully constructed once a path is found to a state with no remaining goals. Figure 3 illustrates an example of a proof tree produced by such a method, alongside the final proof.

The strengths of proof step generation lie in *reusability* and *exploration.* During search, proof states are reusable: if a newly encountered state coincides with one previously explored, they can be merged. Furthermore, the system exhibits strong exploration capabilities by attempting multiple tactics at each step. However, these methods often suffer from slow inference speeds

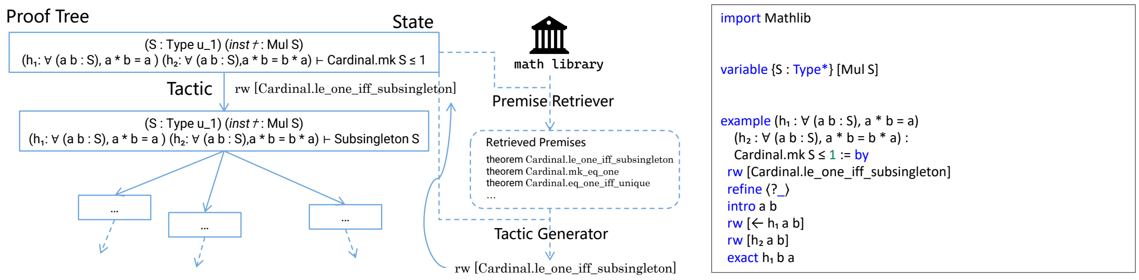

Figure 3: Example of a proof step generation method and the complete proof it produces. Left: An illustration of a proof tree, where each node represents a proof state and each edge corresponds to a tactic. Applying an appropriate tactic transforms the current proof state into the next one. At each step, the proof step generation method retrieves relevant premises and proposes several candidate tactics to try. Right: The complete proof ultimately obtained by this process.

due to the computational cost of tree search, difficulty in training stability, and a heavy reliance on efficient interactive tools for both training and inference.

One of the earliest neural approaches in this domain is *Holophrasm* [161], which employs Monte Carlo Tree Search (MCTS) for exploration. Holophrasm integrates three neural components: a *relevance network* for retrieving useful theorems, a *generative network* for proposing variable substitutions, and a *value network* for estimating provability. Subsequent work largely treated tactic prediction as a classification problem, with representative examples including *GamePad* [76], *DeepHOL* [13], and graph-based approaches [121]. Moving beyond pure classification, *GPT-f* [126] trains a Transformer to generate proof steps via a conditional language modeling objective, constructing proofs using best-first search. Similarly, [95] introduces a hypertree-based search combined with an online training strategy, where policy and critic models are periodically updated on data collected from repeated proof searches.

A major challenge in this domain is the scarcity of large-scale formal data. To mitigate this bottleneck, researchers have pursued two complementary directions: improving model performance under limited formal data, and synthesizing additional formal data. For the former, *ABEL* [65] proposes an online reinforcement learning framework with prioritized sampling to encourage exploration; notably, its RL stage is trained on only 244 formal statements yet achieves performance comparable to the state of the art at the time. For the latter, *REAL-Prover* [138] proposes an integrated pipeline comprising a statement autoformalizer to translate informal statements, a retrieval-augmented prover that generates tactics conditioned on relevant premises, and an *expert iteration* paradigm. In this loop, the model trains on generated state-tactic pairs, performs proof search, and iteratively collects new training data from successful searches. Building on the retrieval-augmented architecture of REAL-Prover, the authors introduced the *reap* [9] tactic as a concrete instantiation of retrieval-guided proof automation. The reap tactic is currently in a beta-testing phase and maintained as a candidate feature branch for mathlib4.

A notable milestone is *AlphaProof* [79]. AlphaProof trains a 3B-parameter proof network that jointly outputs tactics and value estimates. Its training pipeline involves pretraining on 300B tokens, supervised fine-tuning on 300k state-tactic pairs, and reinforcement learning on 80 million autoformalized statements. These formal statements are derived from approximately one million informal problems, where the autoformalization model is trained on triplets of (informal statement, formalization chain-of-thought, formal statement), with multiple distinct translations generated per problem. For particularly challenging tasks, AlphaProof applies *test-time reinforcement learning* to adapt to the problem structure by constructing and training on

[9] https://github.com/frenzymath/reap

a specialized curriculum. As a result, it achieves performance comparable to an IMO silver medalist. Other notable approaches include [83, 125, 150, 172, 167, 104, 168].

Whole-Proof Generation In contrast, whole-proof generation methods aim to produce an entire formal proof, possibly augmented with inline comments, in a single forward pass. Their primary advantages are high inference speed and independence from interactive tools during generation. However, their exploration capability is limited compared to step-wise search; they often rely on behavior cloning and are more susceptible to error accumulation since they lack access to intermediate proof states.

This paradigm relies heavily on data quality and quantity. Since there is no principled a priori method for determining data quality, evaluation is often indirect via model performance. To address data quantity, [165] proposed an integrated pipeline involving automatic statement formalization, filtering (to remove trivial or false statements), statement proving, and iterative training on the resulting verified pairs. Building on this, *DeepSeek-Prover-V1.5* [166] improved performance by constructing richer datasets that include informal proofs written before formal code and inline informal comments, and by applying *Reinforcement Learning from Verifier Feedback (RLVF)*. Other works adopting this paradigm include [56, 12, 174, 47, 180, 148, 106].

Agentic Approaches Agentic approaches [84, 149, 146, 27, 85, 64, 3] represent a paradigm shift from single-model systems to modular workflows. These methods decompose theorem proving into coordinated sub-tasks, such as retrieval, decomposition, and verification, orchestrated through structured workflows that integrate language models with external tools. The effectiveness of these approaches relies on three core components: robust retrieval systems, the reasoning capabilities of LLMs, and workflows designed to mimic the mathematical research process.

Draft, Sketch, and Prove (DSP) [84] is a prototype of this paradigm. It generates an informal proof, translates it into a formal sketch with open subgoals, and closes them using a lightweight prover. *LEGO-Prover* [149] extends this by maintaining a persistent lemma pool. Uniquely, it evolves verified lemmas into new ones through strategies such as *dimension extension*, *key concept identification*, *parameterization*, and *complexity enhancement*. *Hilbert* [146] transforms informal proofs into formal sketches via recursive subgoal decomposition, guided by theorem retrieval. *Seed-Prover-1.5* [27] similarly employs a specialized sketch model and a dedicated prover model, achieving 80% accuracy on the graduate-level benchmark FATE-H and 33% accuracy on the PhD-level benchmark FATE-X [85]. Notably, for agentic approaches such as Seed-Prover, the widely used miniF2F benchmark [183] has largely been saturated and can be regarded as effectively solved. Addressing the granularity gap between informal reasoning and formal code, [157] proposes a two-stage Chain of States (CoS) framework. By extracting a sequence of intermediate formal states aligned with the logical flow of the informal argument before generating the specific transition tactics, this approach significantly reduces the complexity of tactic generation under constrained computational budgets.

Advanced agents like *Aristotle* [3] interleave informal reasoning with formal verification. Aristotle drafts proofs as sequences of lemmas, formalizes them, and attempts verification, refining its outputs iteratively based on feedback. Together with a geometry solver, it achieved performance at the level of an IMO gold medal. *Numina-Lean-Agent* [108] equips the coding agent Claude Code with Lean-LSP-MCP [49] for interacting with Lean, a semantic search engine for theorem retrieval, an informal prover [78], and an external LLM for additional reasoning support. It successfully solved all problems in the 2025 Putnam Competition, an achievement first reported by *AxiomProver* [10]. Numina-Lean-Agent also formalized a research paper on Brascamp–Lieb inequalities [14] by interacting with mathematicians through iterative refinement of a natural language blueprint, represented as a dependency graph for both statements and proofs. Finally, the *Gauss* agent [117] demonstrated the power of human-AI collaboration by formalizing the strong Prime Number Theorem within three weeks, leveraging expert scaffolding. These results demonstrate that carefully designed agentic workflows can effectively exploit both intrinsic reasoning capabilities and external tools to achieve substantial gains in automated theorem proving.

The practical impact of formal reasoning agents is perhaps most clearly illustrated by their emerging role in recent work on conjectures posed by Paul Erdős, as discussed by Terence Tao [10]. During late 2025 and early 2026, agentic systems such as Aristotle became integral components of a high-leverage verification workflow: frontier language models (e.g., Gemini Deep Think, ChatGPT) were used to generate candidate constructions or heuristic arguments, which were then rigorously formalized and mechanically checked in Lean. This pipeline has already yielded formally verified counterexamples to several Erdős problems (notably Problem #205) and partial resolutions of more intricate problems (e.g., #367), while many related cases remain open. Importantly, these examples are best viewed as representative rather than definitive: the process is ongoing, and further results continue to emerge. While formalization dramatically reduces the risk of irreparable hallucinations, it does not eliminate all failure modes—models may still exploit unintended problem specifications or rediscover arguments later found in the literature. Nevertheless, these developments mark a qualitative shift. Formal reasoning agents now function as reliable research copilots, systematically converting heuristic insight into certified mathematical knowledge and mitigating long-standing verification bottlenecks in exploratory mathematics.

3.4 Mathematical Information Retrieval

Mathematical Information Retrieval concerns the task of retrieving mathematical content, including formulas, theorems, and problem solutions, from large collections of mathematical documents. In contrast to standard text retrieval, MIR must explicitly account for the distinctive structure and semantics of mathematical expressions. Mathematical formulas are inherently structured objects whose meaning depends on symbolic composition and relational structure rather than mere lexical overlap. Consequently, an effective MIR system must address challenges such as matching mathematical structure and symbolic patterns, while simultaneously leveraging surrounding textual context to resolve ambiguity and interpret meaning.

Crucially, MIR is not merely a search tool for human users but a foundational component of modern automated theorem proving (ATP) and AI agent systems. In the context of ATP, *premise retrieval*, the task of identifying useful theorems, lemmas, or definitions from a vast library to prove a new theorem, is often the primary bottleneck. As mathematical libraries grow to contain hundreds of thousands of formal statements (e.g., mathlib4), the ability to efficiently retrieve the “needle in the haystack” determines whether a prover can solve a problem or times out. For agentic systems, MIR enables access to long-term mathematical memory, allowing agents to ground their reasoning in established knowledge rather than hallucinating unsupported facts. This requires a shift from traditional keyword-based matching to reasoning-based retrieval. A robust MIR model must understand logical implication and mathematical equivalence; for instance, it must recognize that a theorem stating “a square matrix has a nonzero determinant” is a critical premise required to answer a query about whether “the columns of the matrix form a linearly independent set,” despite the absence of shared keywords between the two statements.

Depending on the granularity of the retrieval target and the nature of the query, MIR encompasses several closely related tasks. Prominent examples include *premise retrieval*, *semantic retrieval*, and *question-answer retrieval*.

[10]https://github.com/teorth/erdosproblems/wiki/AI-contributions-to-Erdős-problems

Premise Retrieval In automated theorem proving, a central subproblem is premise retrieval: given a conjecture and a large library of existing mathematical statements, the system must identify which premises are likely to be useful for constructing a proof. Early approaches relied primarily on handcrafted similarity measures and heuristics[75, 118]. Variants and extensions of these ideas, including tree-based similarity scores, have continued to be explored in more recent work [156]. In parallel, lightweight machine learning methods such as k-nearest neighbors or sparse naive Bayes were also investigated for premise selection [39].

Over the past decade, deep learning methods have increasingly been applied to premise retrieval. A representative early neural approach is *DeepMath* [80], which encodes conjectures and candidate premises separately and the resulting representations are concatenated and passed through a fully connected network that predicts whether a premise is useful for proving the conjecture. Training is supervised using existing proofs, where premises appearing in proofs are treated as positive examples and hard negative mining is employed to construct informative negative samples. Subsequent work sought to better exploit the internal structure of logical formulas. *FormulaNet* [153], for instance, represents each formula as a graph derived from its syntactic structure, with nodes corresponding to constants, variables, or quantifiers. A graph neural network is then used to compute embeddings, which are combined and fed into a classifier to estimate relevance.

Beyond pairwise scoring models, later work explored graph-level representations over entire libraries of statements. In [55], the authors construct a global graph in which nodes correspond to mathematical statements and directed edges encode premise–conclusion relationships extracted from proofs. Premise selection for a new conjecture is formulated as a link prediction problem, and a graph convolutional network is used to score potential edges based on textual and structural features of nodes. In parallel, another line of research adopts embedding-based retrieval methods, treating each mathematical statement as text and encoding it into a single vector using a learned embedding model. Relevance is then assessed via similarity in the embedding space, often followed by a learned re-ranking stage to refine the retrieved candidate set. Training typically relies on contrastive objectives that draw conjectures closer to premises appearing in their proofs while pushing them away from irrelevant statements. Representative examples of this approach include [172, 141, 138].

Semantic Retrieval Semantic retrieval aims to identify mathematically equivalent or closely related statements from a mathematical corpus based on their meaning rather than surface-level similarity. This task is motivated by practical use cases such as theorem search in large mathematical libraries. For example, users of Lean frequently need to locate relevant theorems in mathlib4 when constructing proofs. In this setting, a query may be expressed either in natural language or in formal code, while the retrieval corpus typically consists of formal declarations from mathlib4. To bridge the gap between informal queries and formal corpora, *LeanSearch* [11] constructs an aligned informal–formal corpus derived from mathlib4 and performs retrieval over the combined representation [59]. This approach enables semantic matching across representation modalities and significantly improves retrieval effectiveness for natural language queries. At present, LeanSearch is officially supported by the Lean community and has been integrated into mathlib4 as a standard client for semantic retrieval. This level of community endorsement and infrastructural integration makes LeanSearch a practical and sustainable component of the mathlib ecosystem, rather than a standalone research prototype. In addition to LeanSearch, several other semantic search tools have been developed for mathlib4, including *Moogle* [12], *LeanExplore* [9], *LeanFinder* [111], and *LeanDex* [13].

Beyond formal theorem retrieval, natural-language statement retrieval is also important for

[11] http://leansearch.net/
[12] https://www.moogle.ai/
[13] https://leandex.projectnumina.ai/

mathematical research. It can help mathematicians find related theorems, determine whether a result is already known, avoid redundant effort, and trace the historical origins of a theorem. Existing systems for natural-language semantic search over mathematical statements include *TheoremSearch* [5], which supports semantic search over theorem statements from sources such as arXiv papers, ProofWiki, the Stacks Project, and related mathematical corpora. In *TheoremSearch*, theorem slogans serve as compact textual representations for embedding. Another example is *Matlas* [86], a search engine over 435K published papers and 1.9K textbooks. In *Matlas*, mathematical statements are extracted together with their dependency relations. For each document, these relations are used to construct a directed dependency graph; statements are then unfolded in topological order to obtain more self-contained representations, which are used for embedding and retrieval.

Formula retrieval constitutes an important subtask of semantic retrieval, where the query is a mathematical formula or a formula pattern and the objective is to retrieve semantically related formulas from a collection of documents. This setting introduces unique challenges. Formulas that represent the same mathematical concept may differ significantly in surface form due to notational variation or algebraic properties such as commutativity. Conversely, formulas with similar visual appearance may encode different meanings when interpreted in different mathematical contexts. Traditional approaches to formula retrieval are largely based on tree representations that encode the structural organization of mathematical expressions. In these methods, a formula is represented as a tree, and similarity is defined over trees or their substructures. A widely used representation is the *Symbol Layout Tree (SLT)* [175], in which nodes correspond to symbols and edges encode spatial relationships such as superscript, subscript, or adjacency. Another prominent representation is the *Operator Tree (OPT)* [61], where internal nodes represent operators and leaf nodes represent operands. Compared with SLTs, operator trees abstract away visual layout and focus on mathematical operations and their hierarchical relationships. Tree-based retrieval algorithms typically compare formulas by matching subtrees or paths, or by computing tree edit distances. For example, *Approach0* [186, 185] represents formulas as operator trees and uses leaf-to-root paths as basic retrieval units. It performs retrieval in two stages, first identifying candidate formulas whose paths overlap with those of the query, and then re-ranking candidates based on similarity measures derived from the largest common subtrees.

Beyond traditional symbolic matching, recent work has explored the use of text embedding models for formula retrieval. Early approaches adapt techniques from natural language representation by linearizing structured formula encodings and embedding them in a continuous vector space. For instance, *Tangent-CFT* [115] performs a depth-first traversal over SLT and OPT, tokenizes the resulting tuple sequences, and applies text embedding models to obtain formula representations. Concurrent work augments formula representations with surrounding textual context to better capture semantic meaning [114, 94]. For example, *MathAMR* [114] integrates formulas into their linguistic context by combining Abstract Meaning Representation (AMR) graphs with OPTs, replacing formula nodes in the AMR graph by the root nodes of the corresponding OPTs, and embedding the resulting linearized graphs using Sentence-BERT.

Question-Answer Retrieval Question–answer (QA) retrieval concerns the task of retrieving mathematical answers, explanations, or supporting documents in response to a natural language query. Mathematical questions are inherently multimodal, typically combining natural language with symbolic expressions, formulas, or diagrams, and candidate answers exhibit similar structure. As a result, relevance in mathematical QA retrieval is defined by semantic adequacy, namely whether an answer correctly and meaningfully addresses the question, for example by providing a valid solution, proof, or conceptual explanation, rather than by surface-level lexical overlap.

Traditional general-purpose text retrieval methods, such as BM25 [128, 129] and TF–IDF, rely on sparse vector representations of queries and documents and assess relevance through weighted vector similarity [131]. While these methods can be directly applied to question–answer retrieval, they often perform poorly in mathematical settings. This limitation stems from their dependence on exact term matching and their inability to capture the semantics of mathematical language or the structural relationships encoded in mathematical formulas. With the rise of deep learning, research shifted toward neural retrieval models based on pretrained transformers. A common approach is to pretrain and fine-tune transformer models on large-scale mathematical corpora to obtain representations better aligned with mathematical syntax and semantics. For example, *MathBERT* [123] was pretrained on math-heavy corpora containing formulas and introduced objectives such as masked formula substructure prediction to better model mathematical symbols in context. Building on dense retrieval paradigms, [127] investigated the use of *ColBERT* [89] on the *ARQMath* benchmarks [116, 176], fine-tuning neural retrievers on millions of question–answer pairs with negatives selected via rule-based heuristics. Recognizing the complementary strengths of symbolic and neural methods, several hybrid approaches have also been proposed. *Mabowdor* [184], for example, combines dense passage retrieval with parallel sparse retrieval based on structure-aware mathematical indexing and traditional text-based methods, merging their outputs via a learned weighted scheme. This hybrid strategy achieved strong performance in ARQMath-3 [113], highlighting the effectiveness of integrating classical mathematical structure with neural semantic representations for QA retrieval.

4 Challenges and Prospects

Despite the promising progress in AI for Mathematics, the field faces a fundamental hurdle: current AI systems, particularly foundation models, still lack the depth of reasoning required for research-level mathematics. Bridging this gap necessitates a shift from passive assistance to active learning within a rigorous "logic environment." This requires accelerating the formalization, or digitalization, of mathematics to provide automated, verifiable feedback that can iteratively strengthen AI reasoning. Furthermore, advancing these capabilities demands the infusion of professional mathematical knowledge, ranging from high-quality data curation to the design of specialized agentic workflows, into the model development process. Ultimately, the goal is to integrate AI seamlessly into the daily practice of mathematicians, a vision that can only be realized through continuous, close collaboration between AI researchers, engineers, and the mathematical community. We summarize these key challenges and future directions below:

1. **Domain Expertise and Feature Engineering:** In problem-specific modeling, the design of input features often necessitates deep domain expertise. Human intuition remains indispensable, both for selecting mathematically meaningful features and for interpreting model outputs to derive theoretical insights. This dependency holds equally for discovery-oriented agents (e.g., AlphaEvolve-style systems), which rely on carefully handcrafted representations and scoring functions. Consequently, the development of effective AI for Mathematics requires close and sustained collaboration between machine learning researchers and domain experts to ensure that computational outcomes translate into genuine mathematical progress.

2. **The Verification Bottleneck and Autoformalization:** Accurate and efficient verification constitutes a critical bottleneck in research-level mathematics. The inherent ambiguity of natural language, combined with the scarcity of experts capable of auditing advanced proofs, makes manual verification slow and error-prone. Grounding mathematical reasoning in formal languages, where correctness can be mechanically verified, is often proposed as a means to ensure the reliability of verification. However, the formal reasoning capabilities of current LLMs lag behind their natural language performance due to the severe scarcity of high-quality formal data. Addressing this "formal data gap"

requires the development of robust autoformalization tools that bridge informal and formal mathematics. By building sound infrastructure for specific subfields and supporting repository-level formalization, we can accelerate the translation of natural language reasoning into formal proofs. This creates a virtuous cycle: formally verifiable feedback can serve as a high-quality training signal to further enhance the reasoning capabilities of LLMs in both mathematics and broader domains. Nevertheless, it is important to recognize that informal verification continues to play a valuable role in the verification of research-level mathematics. The *Aletheia* [54] math research agent, for instance, demonstrates how informal verification, through a combination of natural language generation, verification, and revision, can autonomously address mathematical research problems and produce publishable results, highlighting the ongoing relevance of informal verification within the field.

3. **Semantic Consistency in Formalization:** A subtle challenge in autoformalization is verifying the *semantic correctness* of generated formal statements, namely whether the formalized statement faithfully reflects the intent of the original informal statement. Existing models often struggle to determine whether a back-translated formal statement faithfully captures the nuance of the original informal statement. This necessitates the development of fine-grained, robust semantic-consistency verifiers. While the final judgment of semantic intent must arguably remain with human experts to ensure conceptual accuracy, automated systems can serve as efficient first-stage filters. By reducing the volume of candidates that require manual review, such systems can scale the formalization process without compromising rigorous standards.

4. **Beyond Correctness to Understanding:** Formal validity is a necessary, but not sufficient, condition for mathematical value. As William Thurston famously noted [38], "mathematics is not about numbers, equations, computations, or algorithms: it is about understanding." A valuable proof does more than establish truth; it provides insight, reveals structure, and offers techniques applicable to other problems. Similarly, Stanislaw Ulam [145] quotes Stefan Banach as saying, "Good mathematicians see analogies between theorems or theories, the very best ones see analogies between analogies." This highlights a broader truth: the quality of a proof lies in its ability to deepen our conceptual grasp of the mathematical landscape. Future AI systems must therefore aim to go beyond mere verification, assisting in the discovery of proofs that reshape our thinking and uncover connections that were previously invisible.

5. **From Heuristics to Expert Routines:** While standalone LLMs serve as powerful reasoning engines, the future of AI4Math lies in agentic systems designed to emulate the sophisticated workflows of professional mathematicians. Research-level mathematics is rarely a linear deduction; it involves a complex, iterative cycle of constructing examples, consulting literature, formulating conjectures, and refining proof strategies based on intermediate failures. Current agents, however, remain largely generic. A critical frontier is to develop architectures that explicitly model these expert "routines", learning to orchestrate tools and strategies in a way that mirrors the cognitive processes of a researcher. This includes the use of "computational sketching"—leveraging code generation not just for formal proofs, but to construct numerical toy-examples or perform symbolic derivations that quickly verify or falsify human intuition. Furthermore, these agents can automate high-value long-tail tasks often neglected by humans, such as proof reorganization, condition weakening, and the semantic retrieval of obscure existing solutions. Ultimately, the goal is not merely to mimic human workflows but to optimize them, creating agents capable of navigating the search space of mathematical ideas with a systematicity and scale that surpasses human capability in attacking challenging problems.

6. **Active Community Participation:** Resonating with the necessity of domain expertise, the advancement of AI reasoning requires the active intervention of mathematicians. Beyond generating formal data, the community must actively explore these systems to develop a collective "mental model" of their capabilities and boundaries. For instance, distinguishing whether models are more adept at algebraic manipulation than geometric topology is crucial for determining where AI can be reliably deployed. This involves not only accelerating the *digitalization* of mathematical knowledge to create verifiable training corpora but also engaging in "adversarial collaboration" to identify logical gaps. By rigorously mapping these strengths and weaknesses, mathematicians can guide the development of models that are not just statistically powerful but mathematically sound.

7. **Embracing AI-Assisted Research:** We must prepare for a cultural shift where AI evolves from a computational tool into a research *copilot*. Developments in late 2025, highlighted by Georgiev, Gómez-Serrano, Tao and Wagner's work [62], underscore this transition. In this work, the authors observed that while these models may still lack true understanding, often "doing impressions of thinking", they have become adept at autonomously discovering mathematical constructions that elude human intuition. Even when models hallucinate or produce flawed reasoning, their ability to generate plausible structural candidates allows them to act as effective co-pilots, guiding researchers toward fruitful avenues of inquiry while leaving the final, rigorous verification to human experts.

   We would argue that even if an AI's divergent reasoning, proposing random or creative variants, has a low probability of correctness, the total research efficiency improves provided the "verification leverage" is high. In many advanced mathematical domains, generating a solution is computationally or cognitively expensive, whereas verifying a candidate solution is comparatively rapid. This asymmetry allows researchers to utilize AI as a high-volume generator of candidate ideas, where the time saved by a single valid insight significantly outweighs the low cost of discarding incorrect suggestions.

   However, realizing this potential requires more than just powerful models; it demands well-engineered, accessible tools. To facilitate high engagement, the barrier to entry must be lowered through robust software design. The ease of use seen in recent frameworks like AlphaEvolve, compared to earlier prototypes, demonstrates that engineering quality is a critical determinant in transitioning these technologies from experimental curiosities to standard instruments of global adoption.

Acknowledgements

The authors would like to thank Gergely Bérczi, Xiaoshan Gao, Amaury Hayat, Xuhua He, Kyu-Hwan Lee, Geordie Williamson, and Wotao Yin for valuable comments and suggestions that helped improve the presentation of this paper. They also thank Guoxiong Gao and Chenyi Li for collecting the PKU Undergraduate Final Exams and PhD Qualifying Exams. The authors further acknowledge Feiyue Ye, Yining Wang, Rui Wang, Youle Fang and Wenhao Xu for their careful assessment and valuable feedback on machine-generated solutions for the PKU Undergraduate Final Exams, as well as Qing Lan, Zekun Chen, Shuzhe Cai, Shengbo Dong and Yutong Wang for their assessment of solutions for the PKU PhD Qualifying Exams. This work is supported in part by the National Key R&D Program of China grant 2024YFA1014000, the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM113), the New Cornerstone Investigator Program, and the project "Planning Research and Formulation on Artificial Intelligence for Science for the 15th Five-Year Plan," commissioned by the Fourth Department of the Ministry of Science and Technology of the People's Republic of China.

References

[1] Mohammed Abouzaid, Andrew J Blumberg, Martin Hairer, Joe Kileel, Tamara G Kolda, Paul D Nelson, Daniel Spielman, Nikhil Srivastava, Rachel Ward, Shmuel Weinberger, et al. First Proof. *arXiv preprint arXiv:2602.05192*, 2026.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Tudor Achim, Alex Best, Alberto Bietti, Kevin Der, Mathïs Fédérico, Sergei Gukov, Daniel Halpern-Leistner, Kirsten Henningsgard, Yury Kudryashov, Alexander Meiburg, et al. Aristotle: IMO-Level Automated Theorem Proving. *arXiv preprint arXiv:2510.01346*, 2025.

[4] Ayush Agrawal, Siddhartha Gadgil, Navin Goyal, Ashvni Narayanan, and Anand Tadipatri. Towards a Mathematics Formalisation Assistant Using Large Language Models. *arXiv preprint arXiv:2211.07524*, 2022.

[5] Luke Alexander, Eric Leonen, Sophie Szeto, Artemii Remizov, Ignacio Tejeda, Jarod Alper, Giovanni Inchiostro, and Vasily Ilin. Semantic search over 9 million mathematical theorems. *arXiv preprint arXiv:2602.05216*, 2026.

[6] Alberto Alfarano, François Charton, and Amaury Hayat. Global Lyapunov Functions: A Long-Standing Open Problem in Mathematics, with Symbolic Transformers. *Advances in Neural Information Processing Systems*, 37:93643–93670, 2024.

[7] Malik Amir, Yang-Hui He, Kyu-Hwan Lee, Thomas Oliver, and Eldar Sultanow. Machine Learning Class Numbers of Real Quadratic Fields. *International Journal of Data Science in the Mathematical Sciences*, 01(02):107–134, 2023.

[8] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving Inverse Problems Using Data-Driven Models. *Acta Numerica*, 28:1–174, 2019.

[9] Justin Asher. LeanExplore: A Search Engine for Lean 4 Declarations. *arXiv preprint arXiv:2506.11085*, 2025.

[10] Axiom Math. From Seeing Why to Checking Everything. `https://axiommath.ai/territory/from-seeing-why-to-checking-everything`, 2025.

[11] Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W Ayers, Dragomir Radev, and Jeremy Avigad. Proofnet: Autoformalizing and Formally Proving Undergraduate-Level Mathematics. *arXiv preprint arXiv:2302.12433*, 2023.

[12] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An Open Language Model for Mathematics. In *The Twelfth International Conference on Learning Representations*, 2024.

[13] Kshitij Bansal, Sarah Loos, Markus Rabe, Christian Szegedy, and Stewart Wilcox. Holist: An Environment for Machine Learning of Higher Order Logic Theorem Proving. In *International Conference on Machine Learning*, pages 454–463. PMLR, 2019.

[14] Timothée Bénard and Weikun He. Effective Brascamp–Lieb Inequalities. *arXiv preprint arXiv:2511.11091*, 2025.

[15] Gergely Bérczi, Honglu Fan, and Mingcong Zeng. An ML Approach to Resolution of Singularities. In *Topological, Algebraic and Geometric Learning Workshops 2023*, pages 469–487. PMLR, 2023.

[16] Per Berglund, Ben Campbell, and Vishnu Jejjala. Machine Learning Kreuzer–Skarke Calabi–Yau Threefolds. *International Journal of Modern Physics A*, 40(28):2550097, 2025.

[17] Per Berglund, Yang-Hui He, Elli Heyes, Edward Hirst, Vishnu Jejjala, and Andre Lukas. New Calabi–Yau Manifolds from Genetic Algorithms. *Physics Letters B*, 850:138504, 2024.

[18] David S. Berman, Yang-Hui He, and Edward Hirst. Machine Learning Calabi-Yau Hypersurfaces. *Phys. Rev. D*, 105:066002, Mar 2022.

[19] Charles Blundell, Lars Buesing, Alex Davies, Petar Veličković, and Geordie Williamson. Towards Combinatorial Invariance for Kazhdan-Lusztig Polynomials. *Representation Theory of the American Mathematical Society*, 26(37):1145–1191, 2022.

[20] Jonathan W Bober, Andrew R Booker, M Lee, and David Lowry-Duda. Murmurations of Modular Forms in the Weight Aspect. *Algebra and Number Theory*, 2025.

[21] Steven L Brunton and J Nathan Kutz. Promising Directions of Machine Learning for Partial Differential Equations. *Nature Computational Science*, 4(7):483–494, 2024.

[22] Jim Bryan, Balázs Elek, Freddie Manners, George Salafatinos, and Ravi Vakil. The Motivic Class of the Space of Genus 0 Maps to the Flag Variety. *arXiv preprint arXiv:2601.07222*, 2026.

[23] Gergely Bérczi, Baran Hashemi, and Jonas Klüver. Flow-Based Extremal Mathematical Structure Discovery, 2026.

[24] Paul-Jean Cahen, Marco Fontana, Sophie Frisch, and Sarah Glaz. Open problems in commutative ring theory. In *Commutative Algebra: Recent Advances in Commutative Rings, Integer-Valued Polynomials, and Polynomial Functions*, pages 353–375. Springer, 2014.

[25] Jonathan Carifio, James Halverson, Dmitri Krioukov, and Brent D Nelson. Machine Learning in the String Landscape. *Journal of High Energy Physics*, 2017(9):1–36, 2017.

[26] François Charton, Jordan S Ellenberg, Adam Zsolt Wagner, and Geordie Williamson. Patternboost: Constructions in Mathematics with a Little Help from AI. *arXiv preprint arXiv:2411.00566*, 2024.

[27] 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, Zeyu Zheng, Tianxi Zhou, and Thomas Hanwen Zhu. Seed-Prover 1.5: Mastering Undergraduate-Level Theorem Proving via Learning from Experience. *arXiv preprint arXiv:2512.17260*, 2025.

[28] Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to Optimize: A Primer and A Benchmark. *Journal of Machine Learning Research*, 23(189):1–59, 2022.

[29] Yuri Chervonyi, Trieu H Trinh, Miroslav Olšák, Xiaomeng Yang, Hoang Nguyen, Marcelo Menegali, Junehyuk Jung, Vikas Verma, Quoc V Le, and Thang Luong. Gold-Medalist Performance in Solving Olympiad Geometry with AlphaGeometry2. *arXiv preprint arXiv:2502.03544*, 2025.

[30] A Chervov, A Soibelman, S Lytkin, I Kiselev, S Fironov, A Lukyanenko, A Dolgorukova, A Ogurtsov, F Petrov, S Krymskii, et al. CayleyPy RL: Pathfinding and Reinforcement Learning on Cayley Graphs. *arXiv preprint arXiv:2502.18663*, 2025.

[31] Alexander Chervov, Kirill Khoruzhii, Nikita Bukhal, Jalal Naghiyev, Vladislav Zamkovoy, Ivan Koltsov, Lyudmila Cheldieva, Arsenii Sychev, Arsenii Lenin, Mark Obozov, et al. A Machine Learning Approach That Beats Large Rubik's Cubes. *arXiv preprint arXiv:2502.13266*, 2025.

[32] Shang-Ching Chou and Xiao-Shan Gao. Automated Reasoning in Geometry. *Handbook of automated reasoning*, 1:707–749, 2001.

[33] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. A Deductive Database Approach to Automated Geometry Theorem Proving and Discovering. *Journal of Automated Reasoning*, 25(3):219–246, 2000.

[34] D. A. Clarke. Martin Davis. A program for Presburger's algorithm. Summaries of talks presented at the Summer Institute for Symbolic Logic, Cornell University, 1957, 2nd edn., Communications Research Division, Institute for Defense Analyses, Princeton, N.J., 1960, pp. 215–223. *Journal of Symbolic Logic*, 31(1):138–138, 1966.

[35] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021.

[36] Katherine M Collins, Albert Q Jiang, Simon Frieder, Lionel Wong, Miri Zilka, Umang Bhatt, Thomas Lukasiewicz, Yuhuai Wu, Joshua B Tenenbaum, William Hart, et al. Evaluating Language Models for Mathematics Through Interactions. *Proceedings of the National Academy of Sciences*, 121(24):e2318124121, 2024.

[37] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

[38] MARIANA COOK. *Mathematicians: An Outer View of the Inner World*. Princeton University Press, 2009.

[39] Łukasz Czajka and Cezary Kaliszyk. Hammer for Coq: Automation for Dependent Type Theory. *Journal of Automated Reasoning*, 61(1):423–453, 2018.

[40] Alex Davies, András Juhász, Marc Lackenby, and Nenad Tomašev. The Signature and Cusp Geometry of Hyperbolic Knots. *Geometry & Topology*, 28(5):2313–2343, 2024.

[41] Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, et al. Advancing Mathematics by Guiding Human Intuition with AI. *Nature*, 600(7887):70–74, 2021.

[42] Martin Davis. The Prehistory and Early History of Automated Deduction. *Automation of Reasoning 1: Classical Papers on Computational Logic 1957-1966*, pages 1–28, 1983.

[43] Leonardo de Moura and Sebastian Ullrich. The Lean 4 Theorem Prover and Programming Language. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction - CADE 28 - 28th International Conference on Automated Deduction, Virtual Event, July 12-15, 2021, Proceedings*, volume 12699 of *Lecture Notes in Computer Science*, pages 625–635. Springer, 2021.

[44] Leonardo Mendonça de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. The Lean Theorem Prover (System Description). In Amy P. Felty and Aart Middeldorp, editors, *Automated Deduction - CADE-25 - 25th International Conference on Automated Deduction, Berlin, Germany, August 1-7, 2015, Proceedings*, volume 9195 of *Lecture Notes in Computer Science*, pages 378–388. Springer, 2015.

[45] Bin Dong. On Mathematical Modeling in Image Reconstruction and Beyond. In *Proc. Int. Cong. Math*, volume 7, pages 5420–5449, 2022.

[46] Bin Dong, Xuhua He, Pengfei Jin, Felix Schremmer, and Qingchao Yu. Machine Learning Assisted Exploration for Affine Deligne–Lusztig Varieties. *Peking Mathematical Journal*, pages 1–50, 2024.

[47] Kefan Dong and Tengyu Ma. STP: Self-Play LLM Theorem Provers with Iterative Conjecturing and Proving. In *Forty-second International Conference on Machine Learning*, 2025.

[48] Michael R Douglas and Kit Fraser-Taliente. Diffusion Models for Cayley Graphs. *arXiv preprint arXiv:2503.05558*, 2025.

[49] Oliver Dressler. Lean LSP MCP: Tools for Agentic Interaction with the Lean Theorem Prover. `https://github.com/oOo0oOo/lean-lsp-mcp`, 2025.

[50] Boyan Duan, Xiao Liang, Shuai Lu, Yaoxiang Wang, Yelong Shen, Kai-Wei Chang, Ying Nian Wu, Mao Yang, Weizhu Chen, and Yeyun Gong. Gold-Medal-Level Olympiad Geometry Solving with Efficient Heuristic Auxiliary Constructions. *arXiv preprint arXiv:2512.00097*, 2025.

[51] Harold Erbin and Riccardo Finotello. Machine Learning for Complete Intersection Calabi-Yau Manifolds: A Methodological Study. *Phys. Rev. D*, 103:126014, Jun 2021.

[52] Tony Feng. Eigenweights for arithmetic hirzebruch proportionality. *arXiv preprint arXiv:2601.23245*, 2026.

[53] Tony Feng, Junehyuk Jung, Sang-hyun Kim, Carlo Pagano, Sergei Gukov, Chiang-Chiang Tsai, David Woodruff, Adel Javanmard, Aryan Mokhtari, Dawsen Hwang, et al. Aletheia tackles FirstProof autonomously. *arXiv preprint arXiv:2602.21201*, 2026.

[54] Tony Feng, Trieu H Trinh, Garrett Bingham, Dawsen Hwang, Yuri Chervonyi, Junehyuk Jung, Joonkyung Lee, Carlo Pagano, Sang-hyun Kim, Federico Pasqualotto, et al. Towards Autonomous Mathematics Research. *arXiv preprint arXiv:2602.10177*, 2026.

[55] Deborah Ferreira and André Freitas. Premise Selection in Natural Language Mathematical Texts. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7365–7374. Association for Computational Linguistics, 2020.

[56] Emily First, Markus N Rabe, Talia Ringer, and Yuriy Brun. Baldur: Whole-Proof Generation and Repair with Large Language Models. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1229–1241, 2023.

[57] Simon Frieder, Luca Pinchetti, Chevalier Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. Mathematical Capabilities of ChatGPT. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 27699–27744. Curran Associates, Inc., 2023.

[58] Siddhartha Gadgil, Anand Rao Tadipatri, Ayush Agrawal, Ashvni Narayanan, and Navin Goyal. Towards Automating Formalisation of Theorem Statements Using Large Language Models. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022) Workshop on MATH-AI*, 2022.

[59] Guoxiong Gao, Haocheng Ju, Jiedong Jiang, Zihan Qin, and Bin Dong. A Semantic Search Engine for Mathlib4. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 8001–8013, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[60] Guoxiong Gao, Yutong Wang, Jiedong Jiang, Qi Gao, Zihan Qin, Tianyi Xu, and Bin Dong. Herald: A Natural Language Annotated Lean 4 Dataset. In *The Thirteenth International Conference on Learning Representations*, 2025.

[61] Liangcai Gao, Ke Yuan, Yuehan Wang, Zhuoren Jiang, and Zhi Tang. The Math Retrieval System of ICST for NTCIR-12 MathIR Task. In Noriko Kando, Tetsuya Sakai, and Mark Sanderson, editors, *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies, National Center of Sciences, Tokyo, Japan, June 7-10, 2016*. National Institute of Informatics (NII), 2016.

[62] Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao, and Adam Zsolt Wagner. Mathematical Exploration and Discovery at Scale. *arXiv preprint arXiv:2511.02864*, 2025.

[63] Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, et al. Frontiermath: A Benchmark for Evaluating Advanced Mathematical Reasoning in AI. *arXiv preprint arXiv:2411.04872*, 2024.

[64] Fabian Gloeckle, Alex Gu, Gabriel Synnaeve, and Amaury Hayat. Reinforcement Learning for Hierarchical Proof Generation in Lean 4. In *The 5th Workshop on Mathematical Reasoning and AI at NeurIPS 2025*, 2025.

[65] Fabian Gloeckle, Jannis Limperg, Gabriel Synnaeve, and Amaury Hayat. ABEL: Sample Efficient Online Reinforcement Learning for Neural Theorem Proving. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024.

[66] Kurt Gödel. On Formally Undecidable Propositions of Principia Mathematica and Related Systems. 1931.

[67] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Minlie Huang, Nan Duan, Weizhu Chen, et al. ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving. In *The Twelfth International Conference on Learning Representations*.

[68] Sergei Gukov, James Halverson, Ciprian Manolescu, and Fabian Ruehle. Searching for Ribbons with Machine Learning. *Machine Learning: Science and Technology*, 6(2):025065, jun 2025.

[69] Sergei Gukov and Rak-Kyeong Seong. Machine Learning BPS Spectra and the Gap Conjecture. *Phys. Rev. D*, 110:046016, Aug 2024.

[70] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 Incentivizes Reasoning in LLMs Through Reinforcement Learning. *Nature*, 645(8081):633–638, 2025.

[71] Baran Hashemi, Roderic Guigo Corominas, and Alessandro Giacchetto. Can Transformers Do Enumerative Geometry? In *The Thirteenth International Conference on Learning Representations*, 2025.

[72] Yang-Hui He, Kyu-Hwan Lee, Thomas Oliver, and Alexey Pozdnyakov. Murmurations of Elliptic Curves. *Experimental Mathematics*, 34(3):528–540, 2025.

[73] Yiming He, Jia Zou, Xiaokai Zhang, Na Zhu, and Tuo Leng. FGeo-TP: A Language Model-Enhanced Solver for Euclidean Geometry Problems. *Symmetry*, 16(4):421, 2024.

[74] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. *NeurIPS*, 2021.

[75] Kryštof Hoder and Andrei Voronkov. Sine Qua Non for Large Theory Reasoning. In *International Conference on Automated Deduction*, pages 299–314. Springer, 2011.

[76] Daniel Huang, Prafulla Dhariwal, Dawn Song, and Ilya Sutskever. Gamepad: A Learning Environment for Theorem Proving. *arXiv preprint arXiv:1806.00608*, 2018.

[77] Yanxing Huang, Xinling Jin, Sijie Liang, Fuwen Luo, Peng Li, and Yang Liu. FormaRL: Enhancing Autoformalization with No Labeled Data. In *Second Conference on Language Modeling*, 2025.

[78] Yichen Huang and Lin F Yang. Winning Gold at IMO 2025 with a Model-Agnostic Verification-and-Refinement Pipeline. *arXiv preprint arXiv:2507.15855*, 2025.

[79] Thomas Hubert, Rishi Mehta, Laurent Sartran, Miklós Z Horváth, Goran Žužić, Eric Wieser, Aja Huang, Julian Schrittwieser, Yannick Schroecker, Hussain Masoom, et al. Olympiad-Level Formal Mathematical Reasoning with Reinforcement Learning. *Nature*, pages 1–3, 2025.

[80] Geoffrey Irving, Christian Szegedy, Alexander A. Alemi, Niklas Eén, François Chollet, and Josef Urban. DeepMath: Deep Sequence Models for Premise Selection. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016.

[81] Uijeong Jang and Ernest K Ryu. Point Convergence of Nesterov's Accelerated Gradient Method: An AI-Assisted Proof. *arXiv preprint arXiv:2510.23513*, 2025.

[82] Albert Q Jiang, Wenda Li, and Mateja Jamnik. Multilingual Mathematical Autoformalization. *arXiv preprint arXiv:2311.03755*, 2023.

[83] Albert Qiaochu Jiang, Wenda Li, Szymon Tworkowski, Konrad Czechowski, Tomasz Odrzygóźdź, Piotr Miłoś, Yuhuai Wu, and Mateja Jamnik. Thor: Wielding Hammers to Integrate Language Models and Automated Theorem Provers. *Advances in Neural Information Processing Systems*, 35:8360–8373, 2022.

[84] Albert Qiaochu Jiang, Sean Welleck, Jin Peng Zhou, Timothee Lacroix, Jiacheng Liu, Wenda Li, Mateja Jamnik, Guillaume Lample, and Yuhuai Wu. Draft, Sketch, and Prove: Guiding Formal Theorem Provers with Informal Proofs. In *The Eleventh International Conference on Learning Representations*, 2023.

[85] Jiedong Jiang, Wanyi He, Yuefeng Wang, Guoxiong Gao, Yongle Hu, Jingting Wang, Nailing Guan, Peihao Wu, Chunbo Dai, Liang Xiao, et al. FATE: A Formal Benchmark Series for Frontier Algebra of Multiple Difficulty Levels. *arXiv preprint arXiv:2511.02872*, 2025.

[86] Haocheng Ju, Leheng Chen, Peihao Wu, Bryan Dai, and Bin Dong. Matlas: A Semantic Search Engine for Mathematics. *arXiv preprint arXiv:2604.17484*, 2026.

[87] Haocheng Ju, Guoxiong Gao, Jiedong Jiang, Bin Wu, Zeming Sun, Leheng Chen, Yutong Wang, Yuefeng Wang, Zichen Wang, Wanyi He, et al. Automated Conjecture Resolution with Formal Verification. *arXiv preprint arXiv:2604.03789*, 2026.

[88] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-Informed Machine Learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[89] Omar Khattab and Matei Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 39–48, 2020.

[90] Daniel Klaewer and Lorenz Schlechter. Machine Learning Line Bundle Cohomologies of Hypersurfaces in Toric Varieties. *Physics Letters B*, 789:438–443, 2019.

[91] Alexander Koldobsky. Intersection Bodies in R4. *Advances in Mathematics*, 136(1):1–14, 1998.

[92] Alexander Koldobsky. Intersection Bodies, Positive Definite Distributions, and the Busemann-Petty Problem. *American journal of mathematics*, 120(4):827–840, 1998.

[93] János Kollár. Non-Quasi-Projective Moduli Spaces. *Annals of mathematics*, pages 1077–1096, 2006.

[94] Kriste Krstovski and David M Blei. Equation Embeddings. *arXiv preprint arXiv:1803.09123*, 2018.

[95] Guillaume Lample, Timothee Lacroix, Marie-Anne Lachaux, Aurelien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. Hypertree Proof Search for Neural Theorem Proving. *Advances in neural information processing systems*, 35:26337–26349, 2022.

[96] Thomas Lanard and Alberto Mínguez. An Algorithm for Aubert–Zelevinsky Duality\a la M {\oe} glin–Waldspurger. *arXiv preprint arXiv:2509.13231*, 2025.

[97] Robert Tjarko Lange, Yuki Imajuku, and Edoardo Cetin. Shinkaevolve: Towards Open-Ended and Sample-Efficient Program Evolution. *arXiv preprint arXiv:2509.19349*, 2025.

[98] Joonkyung Lee and Jaehyeon Seo. Lower bounds for multivariate independence polynomials and their generalisations. *arXiv preprint arXiv:2602.02450*, 2026.

[99] Kyu-Hwan Lee. Data-Scientific Study of Kronecker Coefficients. *Experimental Mathematics*, 0(0):1–14, 2025.

[100] Kyu-Hwan Lee and Seewoo Lee. Machines Learn Number Fields, But How? The Case of Galois Groups. *arXiv preprint arXiv:2508.06670*, 2025.

[101] Kyu-Hwan Lee, Thomas Oliver, and Alexey Pozdnyakov. Murmurations of Dirichlet Characters. *International Mathematics Research Notices*, 2025(1):rnae277, 2025.

[102] Chenyi Li, Ziyu Wang, Wanyi He, Yuxuan Wu, Shengyang Xu, and Zaiwen Wen. Formalization of Convergence Rates of Four First-Order Algorithms for Convex Optimization. *Journal of Automated Reasoning*, 69(4):28, 2025.

[103] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The Largest Public Dataset in Ai4maths With 860k Pairs of Competition Math Problems and Solutions. *Hugging Face repository*, 13(9):9, 2024.

[104] Yang Li, Dong Du, Linfeng Song, Chen Li, Weikang Wang, Tao Yang, and Haitao Mi. HunyuanProver: A Scalable Data Synthesis Framework and Guided Tree Search for Automated Theorem Proving. *arXiv preprint arXiv:2412.20735*, 2024.

[105] Zhaoyu Li, Jialiang Sun, Logan Murphy, Qidong Su, Zenan Li, Xian Zhang, Kaiyu Yang, and Xujie Si. A Survey on Deep Learning for Theorem Proving. In *First Conference on Language Modeling*, 2024.

[106] Yong Lin, Shange Tang, Bohan Lyu, Jiayun Wu, Hongzhou Lin, Kaiyu Yang, Jia Li, Mengzhou Xia, Danqi Chen, Sanjeev Arora, et al. Goedel-Prover: A Frontier Model for Open-Source Automated Theorem Proving. *arXiv preprint arXiv:2502.07640*, 2025.

[107] Gang Liu, Yihan Zhu, Jie Chen, and Meng Jiang. Scientific Algorithm Discovery by Augmenting Alphaevolve with Deep Research. *arXiv preprint arXiv:2510.06056*, 2025.

[108] Junqi Liu, Zihao Zhou, Zekai Zhu, Marco Dos Santos, Weikun He, Jiawei Liu, Ran Wang, Yunzhou Xie, Junqiao Zhao, Qiufeng Wang, et al. Numina-Lean-Agent: An Open and General Agentic Reasoning System for Formal Mathematics. *arXiv preprint arXiv:2601.14027*, 2026.

[109] Qi Liu, Xinhao Zheng, Xudong Lu, Qinxiang Cao, and Junchi Yan. Rethinking and Improving Autoformalization: Towards a Faithful Metric and a Dependency Retrieval-Based Approach. In *The Thirteenth International Conference on Learning Representations*, 2025.

[110] Xiaoyang Liu, Tao Zhu, Zineng Dong, Yuntian Liu, Qingfeng Guo, Zhaoxuan Liu, Yu Chen, and Tao Luo. ASSESS: A Semantic and Structural Evaluation Framework for Statement Similarity. *arXiv preprint arXiv:2509.22246*, 2025.

[111] Jialin Lu, Kye Emond, Weiran Sun, and Wuyang Chen. Lean Finder: Semantic Search for Mathlib That Understands User Intents. In *2nd AI for Math Workshop @ ICML 2025*, 2025.

[112] Jianqiao Lu, Yingjia Wan, Zhengying Liu, Yinya Huang, Jing Xiong, Chengwu Liu, Jianhao Shen, Hui Jin, Jipeng Zhang, Haiming Wang, et al. Process-Driven Autoformalization in Lean 4. *arXiv preprint arXiv:2406.01940*, 2024.

[113] Behrooz Mansouri, Vít Novotný, Anurag Agarwal, Douglas W. Oard, and Richard Zanibbi. Overview of ARQMath-3 (2022): Third CLEF Lab on Answer Retrieval for Questions on Math (Working Notes). In *Proceedings of CLEF 2022 Working Notes (Conference and Labs of the Evaluation Forum)*, 2022.

[114] Behrooz Mansouri, Douglas W. Oard, and Richard Zanibbi. Contextualized Formula Search Using Math Abstract Meaning Representation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 4329–4333, New York, NY, USA, 2022. Association for Computing Machinery.

[115] Behrooz Mansouri, Shaurya Rohatgi, Douglas W. Oard, Jian Wu, C. Lee Giles, and Richard Zanibbi. Tangent-CFT: An Embedding Model for Mathematical Formulas. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR)*, pages 11–18. ACM, 2019.

[116] Behrooz Mansouri, Richard Zanibbi, Douglas W. Oard, and Anurag Agarwal. Overview of ARQMath-2 (2021): Second CLEF Lab on Answer Retrieval for Questions on Math. In K. Selçuk Candan, Bogdan Ionescu, Lorraine Goeuriot, Birger Larsen, Henning Müller, Alexis Joly, Maria Maistro, Florina Piroi, Guglielmo Faggioli, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 215–238, Cham, 2021. Springer International Publishing.

[117] Math, Inc. Introducing Gauss, an Agent for Autoformalization. `https://www.math.inc/gauss`, 2025.

[118] Jia Meng and Lawrence C Paulson. Lightweight Relevance Filtering for Machine-Generated Resolution Problems. *Journal of Applied Logic*, 7(1):41–57, 2009.

[119] A. Newell and H. Simon. The logic theory machine–A complex information processing system. *IRE Transactions on Information Theory*, 2(3):61–79, 1956.

[120] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, et al. AlphaEvolve: A Coding Agent for Scientific and Algorithmic Discovery. *arXiv preprint arXiv:2506.13131*, 2025.

[121] Aditya Paliwal, Sarah Loos, Markus Rabe, Kshitij Bansal, and Christian Szegedy. Graph Representations for Higher-Order Logic and Theorem Proving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2967–2974, 2020.

[122] Anand Patel. The simplicity of the hodge bundle. *arXiv preprint arXiv:2603.19052*, 2026.

[123] Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. MathBERT: A Pre-trained Model for Mathematical Formula Understanding. *arXiv preprint arXiv:2105.00377*, 2021.

[124] Auguste Poiroux, Gail Weiss, Viktor Kunčak, and Antoine Bosselut. Reliable Evaluation and Benchmarks for Statement Autoformalization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17958–17980, 2025.

[125] Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. Formal Mathematics Statement Curriculum Learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[126] Stanislas Polu and Ilya Sutskever. Generative Language Modeling for Automated Theorem Proving. *arXiv preprint arXiv:2009.03393*, 2020.

[127] Anja Reusch, Maik Thiele, and Wolfgang Lehner. TU_DBS in the ARQMath Lab 2021, CLEF. In *Working Notes of CLEF 2021*, pages 107–124, 2021.

[128] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. Gaithersburg, MD: NIST, January 1995.

[129] Stephen Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, apr 2009.

[130] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M Pawan Kumar, Emilien Dupont, Francisco JR Ruiz, Jordan S Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical Discoveries from Program Search with Large Language Models. *Nature*, 625(7995):468–475, 2024.

[131] Gerard Salton and Christopher Buckley. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[132] Georg Schumacher and Hajime Tsuji. Quasi-Projectivity of Moduli Spaces of Polarized Varieties. *Annals of mathematics*, pages 597–639, 2004.

[133] Georg Schumacher and Hajime Tsuji. Retraction: "Quasi-Projectivity of Moduli Spaces of Polarized Varieties". *Annals of Mathematics*, 198(3):1305–1305, 2023.

[134] Michael Shalyt, Uri Seligmann, Itay Beit Halachmi, Ofir David, Rotem Elimelech, and Ido Kaminer. Unsupervised Discovery of Formulas for Mathematical Constants. *Advances in Neural Information Processing Systems*, 37:113156–113190, 2024.

[135] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024.

[136] Asankhaya Sharma. OpenEvolve: An Open-Source Evolutionary Coding Agent, 2025.

[137] Ali Shehper, Anibal M. Medina-Mardones, Lucas Fagan, Bartłomiej Lewandowski, Angus Gruen, Yang Qiu, Piotr Kucharski, Zhenghan Wang, and Sergei Gukov. What Makes Math Problems Hard for Reinforcement Learning: A Case Study. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[138] Ziju Shen, Naohao Huang, Fanyi Yang, Yutong Wang, Guoxiong Gao, Tianyi Xu, Jiedong Jiang, Wanyi He, Pu Yang, Mengzhou Sun, et al. REAL-Prover: Retrieval Augmented Lean Prover for Mathematical Reasoning. *arXiv preprint arXiv:2505.20613*, 2025.

[139] Stephen G Simpson. Partial Realizations of Hilbert's Program. *The Journal of Symbolic Logic*, 53(2):349–363, 1988.

[140] Grzegorz Swirszcz, Adam Zsolt Wagner, Geordie Williamson, Sam Blackwell, Bogdan Georgiev, Alex Davies, Ali Eslami, Sebastien Racaniere, Theophane Weber, and Pushmeet Kohli. Advancing Geometry with AI: Multi-Agent Generation of Polytopes. *arXiv preprint arXiv:2502.05199*, 2025.

[141] Yicheng Tao, Haotian Liu, Shanwen Wang, and Hongteng Xu. Learning an Effective Premise Retrieval Model for Efficient Mathematical Formalization. In *Proceedings of the 2nd AI for Math Workshop at ICML 2025*, 2025.

[142] Qwen Team. QwQ-32B: Embracing the Power of Reinforcement Learning, March 2025.

[143] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving Olympiad Geometry without Human Demonstrations. *Nature*, 625(7995):476–482, 2024.

[144] S. Ulam. John von Neumann 1903–1957. *Bulletin of the American Mathematical Society*, 64(3.P2):1 – 49, 1958.

[145] Stanislaw M. Ulam. *Adventures of a Mathematician.* Charles Scribner's Sons, New York, 1976.

[146] Sumanth Varambally, Thomas Voice, Yanchao Sun, Zhifeng Chen, Rose Yu, and Ke Ye. Hilbert: Recursively Building Formal Proofs with Informal Reasoning. *arXiv preprint arXiv:2509.22819*, 2025.

[147] Adam Zsolt Wagner. Constructions in Combinatorics via Neural Networks. *arXiv preprint arXiv:2104.14516*, 2021.

[148] Haiming Wang, Mert Unsal, Xiaohan Lin, Mantas Baksys, Junqi Liu, Marco Dos Santos, Flood Sung, Marina Vinyes, Zhenzhe Ying, Zekai Zhu, et al. Kimina-Prover Preview: Towards Large Formal Reasoning Models with Reinforcement Learning. *arXiv preprint arXiv:2504.11354*, 2025.

[149] Haiming Wang, Huajian Xin, Chuanyang Zheng, Zhengying Liu, Qingxing Cao, Yinya Huang, Jing Xiong, Han Shi, Enze Xie, Jian Yin, Zhenguo Li, and Xiaodan Liang. LEGO-Prover: Neural Theorem Proving with Growing Libraries. In *The Twelfth International Conference on Learning Representations*, 2024.

[150] Haiming Wang, Ye Yuan, Zhengying Liu, Jianhao Shen, Yichun Yin, Jing Xiong, Enze Xie, Han Shi, Yujun Li, Lin Li, Jian Yin, Zhenguo Li, and Xiaodan Liang. DT-solver: Automated theorem proving with dynamic-tree sampling guided by proof-level value function. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12632–12646, Toronto, Canada, July 2023. Association for Computational Linguistics.

[151] Hanyu Wang, Ruohan Xie, Yutong Wang, Guoxiong Gao, Xintao Yu, and Bin Dong. Aria: An Agent For Retrieval and Iterative Auto-Formalization via Dependency Graph. *arXiv preprint arXiv:2510.04520*, 2025.

[152] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A Survey on Large Language Model Based Autonomous Agents. *Frontiers of Computer Science*, 18(6):186345, 2024.

[153] Mingzhe Wang, Yihe Tang, Jian Wang, and Jia Deng. Premise Selection for Theorem Proving by Deep Graph Embedding. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

[154] Qingxiang Wang, Chad Brown, Cezary Kaliszyk, and Josef Urban. Exploration of Neural Machine Translation in Autoformalization of Mathematics in Mizar. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 85–98, 2020.

[155] Qingxiang Wang, Cezary Kaliszyk, and Josef Urban. First Experiments with Neural Translation of Informal to Formal Mathematics. In *International Conference on Intelligent Computer Mathematics*, pages 255–270. Springer, 2018.

[156] Zichen Wang, Anjie Dong, and Zaiwen Wen. Tree-Based Premise Selection for Lean4. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[157] Ziyu Wang, Bowen Yang, Shihao Zhou, Chenyi Li, Yuan Zhang, Bin Dong, and Zaiwen Wen. Translating Informal Proofs into Formal Proofs Using a Chain of States. *arXiv preprint arXiv:2512.10317*, 2025.

[158] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[159] E Weinan et al. The Dawning of a New Era in Applied Mathematics. *Notices of the American Mathematical Society*, 68(4):565–571, 2021.

[160] Wu Wen-Tsun. Basic Principles of Mechanical Theorem Proving in Elementary Geometries. *Journal of automated Reasoning*, 2(3):221–252, 1986.

[161] Daniel Whalen. Holophrasm: A Neural Automated Theorem Prover for Higher-Order Logic. *arXiv preprint arXiv:1608.02644*, 2016.

[162] Wen-tsün Wu. *Mechanical Theorem Proving in Geometries: Basic Principles*. Springer Science & Business Media, 2012.

[163] Wenjun Wu and Xiaoshan Gao. Mathematics Mechanization and Applications After Thirty Years. *Frontiers of Computer Science in China*, 1(1):1–8, 2007.

[164] Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with Large Language Models. *Advances in neural information processing systems*, 35:32353–32368, 2022.

[165] Huajian Xin, Daya Guo, Zhihong Shao, Zhizhou Ren, Qihao Zhu, Bo Liu, Chong Ruan, Wenda Li, and Xiaodan Liang. Deepseek-Prover: Advancing Theorem Proving in LLMs through Large-Scale Synthetic Data. *arXiv preprint arXiv:2405.14333*, 2024.

[166] Huajian Xin, Z.Z. Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, Wenjun Gao, Haowei Zhang, Qihao Zhu, Dejian Yang, Zhibin Gou, Z.F. Wu, Fuli Luo, and Chong Ruan. DeepSeek-Prover-V1.5: Harnessing Proof Assistant Feedback for Reinforcement Learning and Monte-Carlo Tree Search. In *The Thirteenth International Conference on Learning Representations*, 2025.

[167] Ran Xin, Chenguang Xi, Jie Yang, Feng Chen, Hang Wu, Xia Xiao, Yifan Sun, Shen Zheng, and Ming Ding. Bfs-Prover: Scalable Best-First Tree Search for LLM-Based Automatic Theorem Proving. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32588–32599, 2025.

[168] Ran Xin, Zeyu Zheng, Yanchen Nie, Kun Yuan, and Xia Xiao. Scaling Up Multi-Turn Off-Policy RL and Multi-Agent Tree Search for LLM Step-Provers. *arXiv preprint arXiv:2509.06493*, 2025.

[169] Yu Xuejun, Jianyuan Zhong, Zijin Feng, Pengyi Zhai, Roozbeh Yousefzadeh, Wei Chong Ng, Haoxiong Liu, Ziyi Shou, Jing Xiong, Yudong Zhou, et al. Mathesis: Towards Formal Theorem Proving from Natural Languages. *arXiv preprint arXiv:2506.07047*, 2025.

[170] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

[171] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[172] Kaiyu Yang, Aidan M Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[173] Huaiyuan Ying, Zijian Wu, Yihan Geng, Jiayu Wang, Dahua Lin, and Kai Chen. Lean Workbook: A Large-Scale Lean Problem Set Formalized from Natural Language Math Problems. *Advances in Neural Information Processing Systems*, 37:105848–105863, 2024.

[174] Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, et al. Internlm-Math: Open Math Large Language Models Toward Verifiable Reasoning. *arXiv preprint arXiv:2402.06332*, 2024.

[175] Richard Zanibbi and Dorothea Blostein. Recognition and Retrieval of Mathematical Expressions. *Int. J. Document Anal. Recognit.*, 15(4):331–357, 2012.

[176] Richard Zanibbi, Douglas W. Oard, Anurag Agarwal, and Behrooz Mansouri. Overview of ARQMath 2020: CLEF Lab on Answer Retrieval for Questions on Math. In Avi Arampatzis, Evangelos Kanoulas, Theodora Tsikrika, Stefanos Vrochidis, Hideo Joho, Christina Lioma, Carsten Eickhoff, Aurélie Névéol, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 169–193, Cham, 2020. Springer International Publishing.

[177] Chi Zhang, Jiajun Song, Siyu Li, Yitao Liang, Yuxi Ma, Wei Wang, Yixin Zhu, and Song-Chun Zhu. Proposing and Solving Olympiad Geometry with Guided Tree Search. *arXiv preprint arXiv:2412.10673*, 2024.

[178] Gaoyong Zhang. Intersection Bodies and the Busemann-Petty Inequalities in R4. *Annals of Mathematics*, 140(2):331–346, 1994.

[179] Gaoyong Zhang. A Positive Solution to the Busemann-Petty Problem in R4. *Annals of Mathematics*, pages 535–543, 1999.

[180] Jingyuan Zhang, Qi Wang, Xingguang Ji, Yahui Liu, Yang Yue, Fuzheng Zhang, Di Zhang, Guorui Zhou, and Kun Gai. Leanabell-Prover: Posttraining Scaling in Formal Reasoning. *arXiv preprint arXiv:2504.06122*, 2025.

[181] Xiaokai Zhang, Na Zhu, Yiming He, Jia Zou, Qike Huang, Xiaoxiao Jin, Yanjun Guo, Chenyang Mao, Zhe Zhu, Dengfeng Yue, et al. FormalGeo: The First Step toward Human-Like IMO-Level Geometric Automated Reasoning. *arXiv preprint arXiv:2310.18021*, 2023.

[182] Xiaokai Zhang, Na Zhu, Cheng Qin, Yang Li, Zhenbing Zeng, and Tuo Leng. FGeo-HyperGNet: Geometric Problem Solving Integrating Formal Symbolic System and Hypergraph Neural Network. *arXiv preprint arXiv:2402.11461*, 2024.

[183] Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. MiniF2F: A Cross-System Benchmark for Formal Olympiad-Level Mathematics. *arXiv preprint arXiv:2109.00110*, 2021.

[184] Wei Zhong, Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. One Blade for One Purpose: Advancing Math Information Retrieval using Hybrid Search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 141–151, New York, NY, USA, 2023. Association for Computing Machinery.

[185] Wei Zhong, Shaurya Rohatgi, Jian Wu, C. Lee Giles, and Richard Zanibbi. Accelerating Substructure Similarity Search for Formula Retrieval. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I*, page 714–727, Berlin, Heidelberg, 2020. Springer-Verlag.

[186] Wei Zhong and Richard Zanibbi. Structural Similarity Search for Formulas Using Leaf-Root Paths in Operator Subtrees. In Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra, editors, *Advances in Information Retrieval - 41st European Conference on IR Research, ECIR 2019, Cologne, Germany, April 14-18, 2019, Proceedings, Part I*, volume 11437 of *Lecture Notes in Computer Science*, pages 116–129. Springer, 2019.

[187] Yichi Zhou, Jianqiu Zhao, Yongxin Zhang, Bohan Wang, Siran Wang, Luoxin Chen, Jiahui Wang, Haowei Chen, Allan Jie, Xinbo Zhang, et al. Solving Formal Math Problems by Decomposition and Iterative Reflection. *arXiv preprint arXiv:2507.15225*, 2025.

[188] Jia Zou, Xiaokai Zhang, Yiming He, Na Zhu, and Tuo Leng. FGeo-DRL: Deductive Reasoning for Geometric Problems through Deep Reinforcement Learning. *Symmetry*, 16(4):437, 2024.

[189] Nina Zubrilina. Murmurations. *Inventiones mathematicae*, 241(3):627–680, 2025.

A Scoring Criteria

Score	Criteria
0	The approach is entirely incorrect.
1	The reasoning is confused and the errors are fatal, but the first one or two lines head in the right direction.
2	The main idea is correct, but errors in the middle cause the entire latter part to be wrong.
3	The main idea is correct; although there are mistakes, the proof eventually returns to the right track and the latter part is correct.
4	The overall proof is correct, with only minor, non-essential errors.
5	Completely correct.

Table 1: Scoring criteria for the undergraduate exams.

B Example Problems from PKU Undergraduate Final Exams

Problem 1. *For integers $1 < p < q$, define $T_{p,q}(x)$ as the following triangular polynomial: $\frac{\cos((q-p)x)}{p} + \frac{\cos((q-p+1)x)}{p-1} + \cdots + \frac{\cos((q-1)x)}{1} - \frac{\cos((q+1)x)}{1} - \frac{\cos((q+2)x)}{2} - \cdots - \frac{\cos((q+p)x)}{p}$ For $k \geq 1$, choose integers $1 < p_k < q_k$ and real numbers $a_k > 0$ such that $q_k + p_k < q_{k+1} - p_{k+1}$, $\sum_{k=1}^{\infty} a_k < +\infty$, and $\liminf_{k\to\infty} a_k \ln p_k > 0$. (Numbers satisfying the above requirements exist, such as $p_k = 2^{k^3}$, $q_k = 2^{k^3+1}$, $a_k = \frac{1}{k^2}$) (1) Prove that $f(x) = \sum_{k=1}^{\infty} a_k T_{p_k,q_k}(x)$ is a continuous function with period 2π. (2) Does the Fourier series of $f(x)$ converge at $x = 0$? Provide a justification.*

Solution (GPT-4). To tackle this problem, we'll break it down into two main parts as requested: proving the continuity and periodicity of $f(x)$, and then discussing the convergence of its Fourier series at $x = 0$.

Part 1: Continuity and Periodicity of $f(x)$

Continuity

To prove $f(x)$ is continuous, we need to show that the series of functions $\sum_{k=1}^{\infty} a_k T_{p_k,q_k}(x)$ converges uniformly on any closed interval in $\mathbb{R}$, because a uniformly convergent series of continuous functions is continuous.

Each $T_{p_k,q_k}(x)$ is a finite sum of continuous functions ($\cos(nx)$ terms divided by integers), and hence is continuous. The uniform convergence follows from the Weierstrass M-test if we can show that $\sum_{k=1}^{\infty} M_k < \infty$, where M_k is a bound for $|a_k T_{p_k,q_k}(x)|$ for each k.

For any x, the maximum value of $|\cos(nx)|$ is 1. Thus, we can bound each term of $T_{p_k,q_k}(x)$ by its denominator's reciprocal. The sum of these bounds forms a telescoping series, which simplifies to a constant times p_k. Therefore, we can choose $M_k = a_k C p_k$ for some constant C that depends on the structure of $T_{p_k,q_k}(x)$ but not on k. Given $\sum_{k=1}^{\infty} a_k < +\infty$ and p_k grows, but not faster than the decrease of a_k, the series $\sum_{k=1}^{\infty} M_k$ converges. Hence, by the Weierstrass M-test, $f(x)$ is continuous.

Periodicity

The periodicity of $f(x)$ with period 2π follows directly from the periodicity of the cosine function, which has a period of 2π. Since each $T_{p_k,q_k}(x)$ is a sum of cosine functions with integer multiples of x, each $T_{p_k,q_k}(x)$ is periodic with period 2π. Therefore, the sum $f(x)$, being a sum of 2π-periodic functions, is also 2π-periodic.

Part 2: Convergence of the Fourier Series at $x = 0$

The Fourier series of a 2π-periodic function $f(x)$ is given by the sum of its Fourier coefficients times the corresponding cosine and sine functions. At $x = 0$, the sine terms vanish, and we are left with the sum of the Fourier cosine coefficients.

To discuss the convergence of the Fourier series of $f(x)$ at $x = 0$, we need to consider the behavior of its Fourier coefficients. The Fourier coefficients involve integrals of $f(x)$ times sine or cosine functions over a period. Given that $f(x)$ is defined as a sum of functions involving cosines, the Fourier coefficients will reflect the structure of these cosines.

However, the question of convergence at $x = 0$ simplifies to considering the sum of the constant terms in the Fourier expansion of $f(x)$. Since $f(x)$ is continuous and periodic, its Fourier series converges to $f(x)$ at points where $f(x)$ is continuous, including $x = 0$, by Dirichlet's test for the convergence of Fourier series.

In summary, the Fourier series of $f(x)$ does converge at $x = 0$, and it converges to $f(0)$ due to the continuity of $f(x)$ at that point. This conclusion is supported by the general theory of Fourier series, which guarantees convergence at points of continuity. □

Solution (o1-2024-12-17). We wish to construct and study the function

$$f(x) = \sum_{k=1}^{\infty} a_k \, T_{p_k, q_k}(x),$$

where each $T_{p,q}(x)$ is the "triangular polynomial"

$$\begin{aligned} T_{p,q}(x) = & \frac{\cos((q-p)x)}{p} + \frac{\cos((q-p+1)x)}{p-1} + \cdots + \frac{\cos((q-1)x)}{1} \\ & - \Big[\frac{\cos((q+1)x)}{1} + \frac{\cos((q+2)x)}{2} + \cdots + \frac{\cos((q+p)x)}{p}\Big], \end{aligned}$$

and where the integers p_k, q_k and the positive real coefficients a_k satisfy:

1) $1 < p_k < q_k$ for each k,
2) $q_k + p_k < q_{k+1} - p_{k+1}$ (so the "frequency blocks" do not overlap),
3) $\sum_{k=1}^{\infty} a_k < +\infty$,
4) $\liminf_{k\to\infty}(a_k \ln p_k) > 0$.

Part (1). $f(x)$ is continuous and 2π-periodic.

Step 1A. A simpler closed-form for $T_{p,q}(x)$.

Although $T_{p,q}(x)$ was originally given as a sum of cosines with certain $1/r$ coefficients, one sees more structure by rewriting it in the following way. Observe that

$$T_{p,q}(x) = \sum_{r=0}^{p-1} \frac{\cos((q-p+r)x)}{p-r} - \sum_{r=1}^{p} \frac{\cos((q+r)x)}{r}.$$

Make the index-shift $n = q - m$ in the first sum and $n = q + r$ in the second sum. A straightforward check shows that

$$T_{p,q}(x) = \sum_{m=1}^{p} \frac{\cos((q-m)x)}{m} - \sum_{r=1}^{p} \frac{\cos((q+r)x)}{r}.$$

Using the trigonometric identity

$$\cos(A) - \cos(B) = -2\sin\left(\frac{A+B}{2}\right)\sin\left(\frac{A-B}{2}\right),$$

one finds

$$\cos((q-m)x) - \cos((q+m)x) = 2\sin(qx)\sin(mx).$$

Hence

$$T_{p,q}(x) = \sum_{m=1}^{p} \frac{\cos((q-m)x) - \cos((q+m)x)}{m} = 2\sin(qx)\sum_{m=1}^{p} \frac{\sin(mx)}{m}.$$

Thus a much simpler, and very useful, identity is

$$T_{p,q}(x) = 2\sin(qx) \cdot S_p(x), \qquad S_p(x) := \sum_{m=1}^{p} \frac{\sin(mx)}{m}. \tag{$\star$}$$

Since sine is bounded by 1, one immediately gets

$$|T_{p,q}(x)| \leq 2\,|\sin(qx)|\,|S_p(x)| \leq 2\sum_{m=1}^{p} \frac{1}{m}.$$

Recalling that the p-th harmonic number $H_p = 1 + 1/2 + \cdots + 1/p$ is on the order of $\ln p$, one has

$$|T_{p,q}(x)| \leq 2H_p \approx 2(\ln p + \text{constants}). \tag{1}$$

In particular, $T_{p,q}(x)$ is a finite linear combination of cosines, hence is continuous and 2π-periodic.

Step 1B. Pointwise (and even uniform) convergence of $f(x)$.
We define

$$f(x) = \sum_{k=1}^{\infty} a_k\, T_{p_k,q_k}(x).$$

Each $T_{p_k,q_k}(x)$ is 2π-periodic and continuous, so any uniform limit of partial sums would also be continuous and 2π-periodic. One must check carefully that the series actually does converge (and is reasonably well-behaved) despite the factor $\ln p_k$ in the upper bound (1).

The crucial extra condition is that the "frequency blocks" in $T_{p_k,q_k}(x)$ do not overlap, because

$$q_k + p_k < q_{k+1} - p_{k+1}.$$

Indeed, $T_{p_k,q_k}(x)$ uses cosines of integer frequencies n in the range

$$n = q_k - p_k, \ldots, q_k + p_k,$$

whereas $T_{p_{k+1},q_{k+1}}(x)$ uses n in the next interval

$$[q_{k+1} - p_{k+1}, \ldots, q_{k+1} + p_{k+1}],$$

which does not intersect the previous one. Consequently, when we form the partial sums

$$F_n(x) = \sum_{k=1}^{n} a_k\, T_{p_k,q_k}(x),$$

each $F_n(x)$ is simply a finite linear combination of cosines with distinct integer frequencies. In particular, each $F_n(x)$ is a trigonometric polynomial (hence continuous, 2π-periodic), and two different blocks of frequencies do not "interfere" with each other.

One can then show (by a Dirichlet-type test in k, or by exploiting the fact that the partial sums in x of those trigonometric blocks do not reinforce each other) that $F_n(x)$ converges uniformly to a limit function $f(x)$. The simplest high-level reason is:

- The series $\sum a_k$ converges (so its partial sums are bounded).
- Each $T_{p_k,q_k}(x)$ oscillates in k because $\sin(q_k x)$ in $(\star)$ has bounded partial sums in k, etc.

In more elementary terms, one can check that for large indices m, n, the difference

$$\|F_n - F_m\|_\infty = \sup_x \left| \sum_{k=m+1}^{n} a_k T_{p_k,q_k}(x) \right|$$

stays small, thanks to the disjoint frequency support and the boundedness of those partial sums. Hence

$$\sum a_k T_{p_k,q_k}(x)$$

converges uniformly to a continuous, 2π-periodic limit $f(x)$. This completes the proof of Part (1).

Part (2). The Fourier series of $f(x)$ does not converge at $x = 0$.

Here we ask about the usual notion of "Fourier series convergence at a point"—namely, taking the N-th partial sum of the full Fourier series (ordered by frequency $n = 0, 1, 2, \ldots$), and then evaluating at $x = 0$. We claim that these partial sums do not converge to the actual value $f(0)$.

Step 2A. Note that $f(0) = 0$.

Indeed, because $\cos(0) = 1$, we get immediately from the original definition

$$T_{p,q}(0) = \left(\frac{1}{p} + \frac{1}{p-1} + \cdots + \frac{1}{1}\right) - \left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{p}\right) = 0.$$

Thus for each k,

$$T_{p_k,q_k}(0) = 0,$$

and hence

$$f(0) = \sum_{k=1}^{\infty} a_k T_{p_k,q_k}(0) = 0.$$

Step 2B. The partial sums of the Fourier series at 0 "jump" and fail to converge.

Next, consider how the partial sums of the full Fourier expansion of f behave at $x = 0$. Since the frequency blocks

$$[q_k - p_k, \ldots, q_k + p_k]$$

do not overlap for different k, each block's contributions to the Fourier coefficients sum to zero:

$$\sum_{n \in \text{block } k} \big[\text{Fourier coefficient of } \cos(nx)\big] = T_{p_k,q_k}(0) = 0.$$

But the key point is that if we only include *part* of the k-th block in the partial sum, we get a large partial harmonic sum that does not vanish. Concretely:

- If $N \geq q_k + p_k$, then the entire block k is included, and its contribution is $T_{p_k,q_k}(0) = 0$.
- If N is just before the end of block k—e.g. $N = q_k - 1$ or $N = q_k$—then only the frequencies $n = q_k - p_k, \ldots, q_k - 1$ appear. These coefficients add up to something of size $a_k H_{p_k}$ (where $H_{p_k} \sim \ln p_k$).

More precisely, the partial Fourier sum up to $N = q_k - 1$ includes the full blocks for all $j < k$, but only the "left half" of block k, yielding

$$\text{partial sum at } 0 = a_k \left(\frac{1}{1} + \frac{1}{2} + \cdots + \frac{1}{p_k}\right) = a_k H_{p_k},$$

which is about $a_k \ln p_k$. By hypothesis,

$$\liminf_{k\to\infty} a_k \ln p_k > 0,$$

so this partial sum does not go to 0. Indeed, as N "enters" a new block, the partial sum at 0 jumps to about $a_k \ln p_k$. Then, once N passes $q_k + p_k$, that entire k-th block again sums to 0. Thus the partial sums at $x = 0$ keep oscillating between approximately $a_k \ln p_k$ and 0, never settling down.

In particular, the sequence of partial sums at 0 fails to converge to $f(0) = 0$. Hence the Fourier series of f does not converge (in the usual sense) at $x = 0$.

Answer Summary

1) Because the blocks $T_{p_k,q_k}(x)$ have disjoint frequency supports and the coefficients a_k sum to a finite number, one shows (by standard uniform-convergence tests for lacunary trigonometric series) that $\sum a_k T_{p_k,q_k}(x)$ converges uniformly. Each triangular block is continuous and 2π-periodic, so its uniform limit $f(x)$ is also continuous and has period 2π.

2) Nonetheless, the usual partial sums of the Fourier series of f at $x = 0$ do not converge to $f(0)$. Indeed $f(0) = 0$; but whenever the partial sums include only a "partial block," one sees a large jump of size roughly $a_k \ln p_k$, which does not vanish as k grows. Thus the classical Fourier partial sums fail to converge at $x = 0$.

□

Comment: For GPT-4, the solution to the first question is correct, but the answer to the second question is not. In contrast, o1-2024-12-17 gives correct solutions to both questions, and its answer to the second question is particularly good because it observes that if a partial Fourier sum includes only part of the k-th frequency block, it produces a large partial harmonic sum that does not vanish.

Problem 2. *Let $\mathbb{R}^{n\times n}$ denote the set of all $n \times n$ real matrices, and E_{ij} be the $n \times n$ matrix with 1 at position (i, j) and 0 elsewhere. Define Γ_r as the set of all $n \times n$ real matrices with rank r. Let $\phi : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ satisfy $\phi(AB) = \phi(A)\phi(B)$ for all $A, B \in \mathbb{R}^{n\times n}$. Prove: (1) $\forall A, B \in \Gamma_r$, $\operatorname{rank}(\phi(A)) = \operatorname{rank}(\phi(B))$; (2) If $\phi(0) = 0$ and there exists a rank 1 matrix W such that $\phi(W) \neq 0$, then there exists an invertible matrix R such that $\phi(E_{ij}) = RE_{ij}R^{-1}$ holds for all E_{ij}, $i, j = 1, 2, \ldots, n$.*

Solution (GPT-4). To prove the given statements, we will proceed step by step, ensuring that each part of the proof is logically sound and well-explained.

Part 1: $\forall A, B \in \Gamma_r$, $\operatorname{rank}(\varphi(A)) = \operatorname{rank}(\varphi(B))$ Proof:

Given that $\varphi : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ satisfies $\varphi(AB) = \varphi(A)\varphi(B)$ for all $A, B \in \mathbb{R}^{n\times n}$, we need to show that for any two matrices A and B with the same rank r, the rank of $\varphi(A)$ is equal to the rank of $\varphi(B)$.

1. **Rank-nullity theorem:** Recall that for any linear transformation $T : V \to W$, the rank-nullity theorem states that $\dim(V) = \operatorname{rank}(T) + \operatorname{nullity}(T)$. This theorem will be foundational in understanding how transformations affect the rank of matrices.

2. **Rank preservation under similar transformations:** If two matrices A and B are similar, i.e., there exists an invertible matrix P such that $A = PBP^{-1}$, then they have the same rank. This is because similarity transformations do not affect the linearly independent rows or columns of a matrix.

3. **Homomorphism property:** The given condition $\varphi(AB) = \varphi(A)\varphi(B)$ indicates that φ is a homomorphism between the algebraic structures $(\mathbb{R}^{n\times n}, \cdot)$ under matrix multiplication. This property will be crucial in understanding how φ preserves or changes the rank of matrices.

To prove that $\text{rank}(\varphi(A)) = \text{rank}(\varphi(B))$ for all $A, B \in \Gamma_r$, consider the following steps:

- **Step 1:** Since A and B are in Γ_r, they have rank r. This means there exist matrices U and V such that $A = UDV$ and $B = XDY$, where D is a diagonal matrix with r non-zero entries (representing the rank), and U, V, X, Y are invertible matrices. This decomposition is a form of the singular value decomposition (SVD), which exists for any real matrix.
- **Step 2:** Apply φ to both sides of the equations $A = UDV$ and $B = XDY$. Using the homomorphism property, we get $\varphi(A) = \varphi(U)\varphi(D)\varphi(V)$ and $\varphi(B) = \varphi(X)\varphi(D)\varphi(Y)$.
- **Step 3:** Since D has rank r, and both A and B are constructed using D with invertible matrices, it's reasonable to infer (though not directly) that the transformation φ applied to matrices of rank r would yield matrices whose rank is determined by the transformation applied to D. However, without additional properties of φ (e.g., linearity, injectivity), directly concluding that $\text{rank}(\varphi(A)) = \text{rank}(\varphi(B))$ requires further justification.

Gap in Proof for Part 1: The direct leap from the homomorphism property to rank preservation without additional properties of φ (such as it being a linear transformation that preserves rank or injectivity/surjectivity) is a logical gap. To rigorously prove that $\text{rank}(\varphi(A)) = \text{rank}(\varphi(B))$ for all $A, B \in \Gamma_r$, we would need more information about the nature of φ beyond its homomorphism property. For instance, if φ were shown to be a linear transformation that either preserves rank or has a predictable effect on rank, we could make more definitive statements.

Part 2: If $\varphi(0) = 0$ and there exists a rank 1 matrix W such that $\varphi(W) \neq 0$, then there exists an invertible matrix R such that $\varphi(E_{ij}) = RE_{ij}R^{-1}$ holds for all E_{ij}, $i, j = 1, 2, \ldots, n$. Proof:

Given the conditions, we aim to show that φ acts similarly to a conjugation by an invertible matrix R on the basis matrices E_{ij}.

1. **Existence of W:** The existence of a rank 1 matrix W such that $\varphi(W) \neq 0$ suggests that φ does not annihilate all rank 1 matrices. Since E_{ij} are also rank 1 matrices, it's plausible to consider how φ acts on these basis elements.
2. **Conjugation property:** The statement we need to prove essentially says that φ acts on E_{ij} as if it were conjugating them with some fixed invertible matrix R. This would mean that φ preserves the structure of E_{ij} matrices up to a basis change represented by R.

Gap in Proof for Part 2: To establish the existence of such an R, we would typically rely on the structure of φ being such that it preserves linear combinations and acts uniformly on the basis elements of the space of matrices. However, without explicit details on how φ acts on general matrices and without assuming linearity of φ, constructing such an R directly from the given conditions is non-trivial.

Conclusion: The proofs for both parts as presented above contain logical gaps primarily due to the lack of explicit assumptions about the properties of φ beyond its homomorphism property. For a rigorous proof, additional information about φ (e.g., linearity, effects on rank, or specific actions on basis elements) would be necessary.

$\square$

Solution (o1-2024-12-17). We shall prove the two statements in turn. Throughout the proof, let $\varphi\colon \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ be a map (not assumed a priori to be unital) satisfying

$$\varphi(AB) = \varphi(A)\,\varphi(B),$$

for all $A, B \in \mathbb{R}^{n\times n}$. Recall that Γ_r denotes the set of all $n \times n$ real matrices of rank r, and E_{ij} is the matrix having a 1 in the (i, j)-entry and 0 elsewhere.

A key tool in the argument is the fact that $M_n(\mathbb{R})$ (the ring of all $n \times n$ real matrices) is a simple ring (over the field $\mathbb{R}$). Being "simple" means that $M_n(\mathbb{R})$ has no nontrivial two-sided ideals (the only two-sided ideals in $M_n(\mathbb{R})$ are $\{0\}$ and $M_n(\mathbb{R})$ itself).

Given a ring-homomorphism $\varphi\colon M_n(\mathbb{R}) \to M_n(\mathbb{R})$ (where "ring-homomorphism" here means preserving multiplication, but not necessarily preserving the identity unless one checks $\varphi(I)$), one knows from standard ring theory:

- Either φ is the zero map (i.e. $\varphi(A) = 0$ for all A),
- Or φ is injective, because the kernel of φ is a two-sided ideal in a simple ring $M_n(\mathbb{R})$, hence must be either $\{0\}$ or all of $M_n(\mathbb{R})$. If φ is not identically zero, its kernel cannot be all of $M_n(\mathbb{R})$, so its kernel is $\{0\}$, and φ is injective.

We shall use this fact repeatedly. Whenever we refer to "simplicity of $M_n(\mathbb{R})$", we are using precisely that the only two-sided ideals are $\{0\}$ or $M_n(\mathbb{R})$.

Statement (1) says:

"If A and B are both rank-r matrices, then $\varphi(A)$ and $\varphi(B)$ have the same rank."

Equivalently, φ is "constant on ranks": for each $r = 0, 1, \ldots, n$ there is some value (say α_r) so that $\mathrm{rank}(\varphi(A)) = \alpha_r$ for every A of rank r.

STEP 1A. Either φ is the zero map, or else it is injective.

By the simplicity argument above, the kernel of φ is a two-sided ideal of $M_n(\mathbb{R})$. If φ is not identically zero, that kernel must be $\{0\}$, so φ becomes injective.

- *Case (i):* φ is the zero map. Then trivially $\mathrm{rank}(\varphi(A)) = 0$ for all A, so if A, B happen to have rank r, indeed $\mathrm{rank}(\varphi(A)) = \mathrm{rank}(\varphi(B)) = 0$. In this case, statement (1) is true (trivially).
- *Case (ii):* φ is nonzero and hence injective. We shall show that, in this scenario, φ sends any two matrices of the same rank to matrices whose ranks coincide.

STEP 1B. $\varphi(I)$ must be an idempotent, and either $\varphi(I) = 0$ or $\varphi(I)$ acts as an identity in the image.

Because $I \cdot I = I$, applying φ gives

$$\varphi(I)\varphi(I) = \varphi(I).$$

So $\varphi(I)$ is an idempotent. An idempotent E in $M_n(\mathbb{R})$ either has some nonzero rank (possibly $< n$) or can be zero.

- If $\varphi(I) = 0$, then for any matrix A,

$$\varphi(A) = \varphi(AI) = \varphi(A)\varphi(I) = \varphi(A) \cdot 0 = 0.$$

That forces φ to be the zero map on all of $M_n(\mathbb{R})$, contradicting that we are now in the nonzero case. Hence in the nonzero/injective scenario, $\varphi(I)$ cannot be zero.

Therefore $\varphi(I) \neq 0$ is an idempotent in the image ring $\varphi(M_n(\mathbb{R}))$. In this situation (a standard lemma in ring theory) one shows that $\varphi(I)$ must actually serve as the identity element of the subring $\varphi(M_n(\mathbb{R}))$. Concretely, for every A,

$$\varphi(A) = \varphi(AI) = \varphi(A)\varphi(I),$$

so $\varphi(I)$ acts like a multiplicative identity on the right of $\varphi(A)$. A similar argument on the left shows $\varphi(I)$ is a two-sided identity.

Thus in the nonzero case, φ is a "unital homomorphism" onto its image.

STEP 1C. If A is invertible, then $\varphi(A)$ must be invertible (in the subring $\varphi(M_n(\mathbb{R}))$).

Suppose $A \in M_n(\mathbb{R})$ is invertible with inverse A^{-1}. Then

$$AA^{-1} = I,$$

so

$$\varphi(A)\varphi(A^{-1}) = \varphi(I),$$

the identity in the image. Hence $\varphi(A)$ has a two-sided inverse $\varphi(A^{-1})$, so $\varphi(A)$ is invertible.

Consequently, whenever A is an invertible $n \times n$ matrix, the image $\varphi(A)$ also is invertible (and thus has rank n).

STEP 1D. Conclude that matrices of a fixed rank r are sent to matrices all of the same rank.

We now show that any two rank-r matrices A, B get sent to matrices of the same rank under φ.

FACT: Any two rank-r matrices $A, B \in M_n(\mathbb{R})$ can be related via invertible left–right factors. Indeed, if $\operatorname{rank}(A) = r$, one can write

$$A = PDQ,$$

where $P, Q \in GL_n(\mathbb{R})$ and $D = \operatorname{diag}(I_r, 0, \ldots, 0)$. Similarly, $B = P'DQ'$ for some invertible P', Q'. Thus

$$B = (P'P^{-1})\, A\, (Q^{-1}Q').$$

Apply φ to B:

$$\varphi(B) = \varphi(RAS) = \varphi(R)\, \varphi(A)\, \varphi(S).$$

Since (by Step 1C) $\varphi(R)$ and $\varphi(S)$ are invertible (in the image), multiplying $\varphi(A)$ on the left and right by invertibles does not change the rank. Therefore

$$\operatorname{rank}(\varphi(B)) = \operatorname{rank}(\varphi(A)).$$

Hence any two rank-r matrices must have images under φ of the same rank:

$$\forall\, A, B \in \Gamma_r, \qquad \operatorname{rank}(\varphi(A)) = \operatorname{rank}(\varphi(B)).$$

This completes the proof of (1).

Statement (2) says:

Assume additionally $\varphi(0) = 0$ and that there exists a rank-1 matrix W with $\varphi(W) \neq 0$. Then there exists an invertible matrix R such that $\varphi(E_{ij}) = RE_{ij}R^{-1}$ for all i, j.

That is, under those extra conditions, φ is actually an inner automorphism of the matrix algebra.

STEP 2A. φ is nonzero, hence injective and unital as in (1).

Since W has rank 1 and $\varphi(W) \neq 0$, φ is not the zero map. By simplicity, φ is injective, and by Step 1B, φ is unital onto its image.

STEP 2B. Minimal idempotents of rank 1 go to pairwise orthogonal idempotents of rank 1.

The standard E_{ii} each have rank 1. By part (1), all $\varphi(E_{ii})$ have the same rank, and the existence of W forces this rank to be at least 1. Since $\varphi(E_{ii})$ is an idempotent, its rank equals its trace, so all must be rank 1.

If $i \neq j$, then $E_{ii}E_{jj} = 0$, hence $\varphi(E_{ii})\varphi(E_{jj}) = \varphi(0) = 0$, so the rank-1 idempotents are orthogonal.

STEP 2C. Decomposing the identity: $\sum_i E_{ii} = I \implies \sum_i \varphi(E_{ii}) = \varphi(I)$.

Because $\varphi(I)$ is the identity in the image, the sum of the n orthogonal rank-1 idempotents is the n-dimensional identity. Thus $\mathbb{R}^n$ decomposes into n one-dimensional invariant subspaces.

Pick a nonzero v_i in the image of $\varphi(E_{ii})$ for each i. Then $\{v_1, \ldots, v_n\}$ is a basis.

STEP 2D. Define R so that conjugation by R agrees with φ on each E_{ij}.

Let $e_1, \ldots, e_n$ be the standard basis. Define R by

$$R(e_i) = v_i.$$

This is invertible.

We show $\varphi(E_{ij}) = RE_{ij}R^{-1}$.

E_{ij} satisfies $E_{ij}(e_k) = 0$ unless $k = j$, and $E_{ij}(e_j) = e_i$.

Conjugation gives

$$(RE_{ij}R^{-1})(v_k) = RE_{ij}(e_k) = \begin{cases} 0, & k \neq j, \\ v_i, & k = j. \end{cases}$$

Similarly, $\varphi(E_{ij})$ kills all v_k with $k \neq j$, and $\varphi(E_{ij})(v_j)$ lies in the span of v_i, with scalar 1 after normalization.

Thus $\varphi(E_{ij})$ and $RE_{ij}R^{-1}$ act identically on the basis $\{v_1, \ldots, v_n\}$, hence are equal.

This completes the proof of (2).

1) We used that $M_n(\mathbb{R})$ is a simple ring. This gives the dichotomy "either φ is zero or injective".

2) We used that if φ is injective and sends 1 to an idempotent $\varphi(1)$, then either $\varphi(1) = 0$ (forcing φ to be zero) or else $\varphi(1)$ serves as the identity in the image.

3) The step (1C) follows from $AA^{-1} = I \implies \varphi(A)\varphi(A^{-1}) = \varphi(I)$.

Putting all pieces together:

- **Part (1):** For each rank r, all $A \in \Gamma_r$ get mapped to matrices of a single rank.
- **Part (2):** Under $\varphi(0) = 0$ and the existence of W of rank 1 with $\varphi(W) \neq 0$, φ is an inner automorphism:
$$\varphi(E_{ij}) = RE_{ij}R^{-1}.$$

This completes the required proofs.

□

Comment: The GPT-4 solution correctly notes that multiplicativity alone does not immediately imply rank preservation, but it ultimately does not provide proofs of (1) or (2). The o1 solution contains several substantial gaps: it treats $\ker \varphi$ as a two-sided ideal (which requires additivity), uses "simplicity" to deduce injectivity, and later applies additivity when manipulating $\sum_i E_{ii} = I$. Moreover, invertibility in the image algebra with identity $\varphi(I)$ does not automatically imply invertibility in $M_n(\mathbb{R})$, so the rank arguments need further justification.

Problem 3. *Let $P(x)$ be a real polynomial of degree n , which satisfies $\int_a^b P(x)x^k dx = 0, k = 0, \cdots, n-1$. (1)Prove that $P(x)$ has n real simple roots at (a, b). (2)Prove that for an n-point integration formula on the interval $[a, b]$, if the roots of the equation $P(x) = 0$ are used as nodes, the algebraic precision can reach up to $2n - 1$ order.*

Solution (GPT-4). To address this problem, we'll tackle it in two parts as requested.

Part 1: Proving $P(x)$ has n real simple roots in (a, b)

Given that $P(x)$ is a real polynomial of degree n satisfying the condition $\int_a^b P(x)x^k dx = 0$ for $k = 0, \cdots, n-1$, we aim to show that $P(x)$ has n real simple roots in the interval (a, b).

Proof:

1. **Orthogonality Principle**: The condition $\int_a^b P(x)x^k dx = 0$ for $k = 0, \cdots, n-1$ implies that $P(x)$ is orthogonal to all polynomials of degree less than n on the interval $[a, b]$. This is a key observation derived from the properties of inner products in the vector space of polynomials.

2. **Existence of Roots**: Assume for contradiction that $P(x)$ does not have n real roots in (a, b). Since $P(x)$ is a polynomial of degree n, it must have exactly n roots in the complex plane (counting multiplicities) by the Fundamental Theorem of Algebra. If not all of these roots are real and simple within (a, b), then either some roots are complex, or they are real but outside (a, b), or some roots are multiple.

3. **Role of Orthogonality**: If $P(x)$ does not vanish at n distinct points in (a, b), then there exists at least one polynomial $Q(x)$ of degree less than n such that $Q(x)$ does not change sign on (a, b) where $P(x)$ does not vanish. This implies $\int_a^b P(x)Q(x)dx \neq 0$, contradicting the given orthogonality condition.

4. **Conclusion**: Therefore, $P(x)$ must have n real, simple roots in the interval (a, b). This conclusion follows from the contradiction in step 3 and the fact that a polynomial of degree n cannot be orthogonal to a non-zero polynomial of lower degree unless it has specific sign changes that allow the integral to vanish.

Part 2: Proving the Algebraic Precision of an n-point Integration Formula

Given an n-point integration formula on the interval $[a, b]$ using the roots of $P(x) = 0$ as nodes, we need to prove that its algebraic precision can reach up to $2n - 1$ order.

Proof:

1. **Gauss Quadrature Principle**: The n-point Gauss quadrature formula is exact for polynomials of degree up to $2n - 1$. This is because it uses the roots of orthogonal polynomials (with respect to a weight function, which is 1 in our case) as nodes. Our polynomial $P(x)$, satisfying the given conditions, acts similarly to these orthogonal polynomials.

2. **Construction of the Quadrature**: Since the roots of $P(x)$ are used as nodes, and we've established that $P(x)$ has n real, simple roots within (a, b), these nodes are the optimal choice for constructing a quadrature formula with the highest possible degree of precision.

3. **Precision of the Quadrature**: The precision of an integration formula is determined by the highest degree of polynomial it can integrate exactly. By the construction of Gauss quadrature and the properties of orthogonal polynomials, it's known that using the roots of such polynomials as nodes yields a formula that is exact for polynomials of degree up to $2n - 1$.

4. **Conclusion**: Therefore, the n-point integration formula using the roots of $P(x)$ as nodes has an algebraic precision of up to $2n - 1$ order. This is a direct consequence of the properties of orthogonal polynomials and the Gauss quadrature principle.

In summary, the given conditions on $P(x)$ ensure it behaves similarly to orthogonal polynomials used in Gauss quadrature, allowing us to conclude both the existence of n real simple roots within (a, b) and the high degree of precision for the corresponding n-point integration formula.

□

Solution (o1-2024-12-17). We are given a real polynomial $P(x)$ of degree n on the interval $[a, b]$ satisfying the orthogonality conditions

$$\int_a^b P(x)\, x^k\, dx = 0 \quad \text{for } k = 0, 1, 2, \ldots, n-1.$$

There are two main claims to prove:

1) $P(x)$ has n real, simple (i.e., distinct) roots in (a,b). 2) If one uses these n roots as the nodes of an n-point quadrature rule on $[a,b]$, then that rule is exact (i.e., has "algebraic precision") for all polynomials of degree $\leq 2n-1$.

PART (1). $P(x)$ **HAS** n **REAL, SIMPLE ROOTS IN** (a,b)

Step 1A. $P(x)$ must change sign at least n times in (a,b).

- First, observe that $P(x)$ cannot remain strictly nonnegative nor strictly nonpositive on the entire interval (a,b).
  - If $P(x) \geq 0$ for all $x \in (a,b)$, then the integral $\int_a^b P(x)\,dx$ would be strictly positive (unless P is the zero polynomial, which it is not since it has degree n). This contradicts the orthogonality condition for $k=0$:

$$\int_a^b P(x) \cdot 1\,dx = 0.$$

  - A similar argument shows it cannot be strictly nonpositive either.

Hence, $P(x)$ must take on both positive and negative values somewhere in (a,b). By the Intermediate Value Theorem, each sign change requires at least one real root. Next, we show that there must be at least n sign changes in (a,b), forcing at least n roots.

Step 1B. Constructing a contradiction if there are fewer than n roots.

Assume for contradiction that $P(x)$ has at most $n-1$ real roots in (a,b). Then it can have at most $n-1$ sign changes.

We now construct a polynomial $Q(x)$ of degree $< n$ whose integral against $P(x)$ does not vanish. Roughly, we let $Q(x)$ "follow" the sign of $P(x)$ so that $P(x)Q(x)$ is nonnegative or nonpositive and strictly nonzero on large subintervals.

Sketch of the construction of $Q(x)$:

- Partition (a,b) into subintervals where $P(x)$ is positive or negative.
- Pick one point in each subinterval. Call them $x_1,\ldots,x_m$, with $m \leq n-1$.
- Define $Q(x)$ of degree $\leq m-1$ (hence $\leq n-2 < n$) so that

$$Q(x_i) = \begin{cases} +1 & \text{if } P(x_i) > 0, \\ -1 & \text{if } P(x_i) < 0. \end{cases}$$

  Interpolation guarantees this is possible.

Because $Q(x_i)$ has the same sign as $P(x_i)$, on each subinterval the product $P(x)Q(x)$ keeps one sign and does not integrate to zero. Thus

$$\int_a^b P(x)Q(x)\,dx \neq 0,$$

contradicting orthogonality to all polynomials of degree $< n$.

Hence $P(x)$ must have at least n real roots in (a,b).

Step 1C. $P(x)$ must be of exact degree n, so it has exactly n real roots.

A degree-n polynomial has at most n real roots. Combined with at least n roots, we obtain exactly n roots.

Step 1D. The roots are distinct (no multiple roots).

A standard result for orthogonal polynomials states:

- For an orthogonal polynomial $P(x)$ (with respect to a positive weight), the roots of P and P' interlace.

- Thus P and P' cannot share a root unless P is identically zero, so P has no repeated roots.

Alternatively, one may argue directly that a repeated root leads to fewer sign changes or produces a contradiction by constructing an appropriate $Q(x)$.

Thus $P(x)$ has n distinct real roots in (a, b).

PART (2). A $2n-1$ ORDER QUADRATURE RULE AT THE ROOTS

We now prove that if one takes the n distinct real roots of $P(x)$ as the nodes of an n-point formula on $[a, b]$, then this formula is exact for all polynomials of degree up to $2n - 1$.

Recall the statement:

An n-point quadrature rule has algebraic precision m if $\int_a^b R(x)\,dx = \sum_{i=1}^n w_i R(x_i)$ for all $\deg(R) \leq m$.

We will show $m = 2n - 1$ is attainable with nodes x_i equal to the roots of P.

Step 2A. Choosing weights w_i to integrate all polynomials of degree $\leq n - 1$.

Let $x_1, \ldots, x_n$ be the n distinct real roots of $P(x)$. We seek $w_1, \ldots, w_n$ such that

$$\int_a^b R(x)\,dx = \sum_{i=1}^n w_i R(x_i)$$

for every polynomial R of degree $\leq n - 1$.

It suffices to impose this on the basis $1, x, x^2, \ldots, x^{n-1}$. This gives n linear equations for n unknowns w_i. The distinctness of x_i ensures the system is nonsingular (Vandermonde-like), so the solution exists and is unique.

Step 2B. Exactness up to degree $2n - 1$ follows from orthogonality.

Let $Q(x)$ be any polynomial with $\deg(Q) \leq 2n - 1$. Perform polynomial division:

$$Q(x) = A(x) + B(x)P(x),$$

where $\deg(A) < n$ and $\deg(B) \leq n - 1$.

Compute the integral:

$$\int_a^b Q(x)\,dx = \int_a^b A(x)\,dx + \int_a^b B(x)P(x)\,dx.$$

Orthogonality gives

$$\int_a^b B(x)P(x)\,dx = 0,$$

because $\deg(B) \leq n - 1$. Thus

$$\int_a^b Q(x)\,dx = \int_a^b A(x)\,dx.$$

Now compute the quadrature sum:

$$\sum_{i=1}^n w_i Q(x_i) = \sum_{i=1}^n w_i \big[A(x_i) + B(x_i)P(x_i)\big].$$

But $P(x_i) = 0$ for all i, so this equals

$$\sum_{i=1}^n w_i A(x_i).$$

By construction of w_i,

$$\sum_{i=1}^n w_i A(x_i) = \int_a^b A(x)\,dx.$$

Therefore,

$$\sum_{i=1}^{n} w_i Q(x_i) = \int_a^b Q(x)\, dx.$$

Thus the quadrature rule is exact for all polynomials of degree at most $2n-1$.

CONCLUSION

1) The orthogonality conditions $\int_a^b P(x)x^k\, dx = 0$ for $k = 0, \ldots, n-1$ force $P(x)$ to change sign at least n times in (a, b). Since P is degree n, it has exactly n distinct real roots in (a, b).

2) Using these n roots as quadrature nodes and choosing weights w_i appropriately yields a rule exact for all polynomials of degree up to $2n-1$. Any polynomial Q of degree $\leq 2n-1$ decomposes as $Q = A + BP$, orthogonality kills the integral of BP, and the root property kills its quadrature contribution.

This completes the proofs of (1) and (2).

Remark (How theorems are used).

- The Intermediate Value Theorem guarantees real roots when a continuous function changes sign.
- A theorem from orthogonal polynomial theory ensures P and P' cannot share roots, so no repeated roots.
- The argument for degree $2n-1$ exactness is the classical Gaussian quadrature proof using polynomial division and orthogonality.

□

Comment: For GPT-4's solution, the idea for the first question is correct, but the derivation of the contradiction is not explained in sufficient detail. The second question is not addressed at all; the response simply restates the problem without providing a proof. For o1-2024-12-17's solution, the argument in Part (1) correctly uses orthogonality to show that the polynomial must change sign multiple times, leading to the conclusion that it has n real, distinct roots. The proof of Part (2) is also correct.