Mathematisches Forschungsinstitut Oberwolfach

Report No. 43/2025

DOI: 10.4171/OWR/2025/43

MATRIX-MFO Tandem Workshop: Machine Learning and AI for Mathematics

Organized by
François Charton, Paris
Jan de Gier, Melbourne
Amaury Hayat, Paris
Julia Kempe, New York
Geordie Williamson, Sydney

21 September – 27 September 2025

Abstract. This workshop explored how modern machine learning can both accelerate mathematical discovery and preserve rigorous standards. It focused on three angles: using AI techniques to help mathematicians make advances on challenging problems; using mathematics to understand AI predictions; and using deep-learning models for automated theorem proving. Key discussions included using machine learning as a tool for constructing interesting mathematical constructions and navigating in mathematical search spaces, to uncover conjectures and high-quality examples (e.g., sphere packings via DiffuseBoost, combinatorial objects via AlphaEvolve); Integrating Large Language Models (LLMs) with formal systems (e.g., Lean/mathlib) to create scalable, certifiable AI-based automated theorem prover; Collaborative formalization (e.g., the Carleson theorem project), autoformalization for high-quality supervised data, and reinforcement learning/search methods for proof generation and algorithmic reasoning.

Mathematics Subject Classification (2020): 68T07, 03B35, 68V15, 68V20, 68Q32, 68T05, 68Q25.

Introduction by the Organizers

Having intelligent computers able to solve complicated problems on their own has been a sci-fi fantasy for almost as long as computers have existed. The progress of Artificial Intelligence (AI), and in particular Deep Neural Networks, in the last 20 years has made this a reality for a number of tasks and has revolutionized some

areas such as vision [26, 29] or translation and natural language processing [**?**, 25]. While it is usually conceivable that an AI could translate words, play chess, or process data as well or better than humans, it has often been hard to believe that they could perform abstract mathematics on their own. Nevertheless, over the last few years, works on AI for mathematics have been developing at a rapid pace and AI techniques have enabled new discoveries in mathematics [9, 12, 31] in knot theory [21, 9, 15], representation theory [3], partial differential equations [28], dynamical systems [1], control theory [18, 2] and many others. Interestingly, in many cases the neural networks involved are relatively small, far from large language models. This may indicate that neural networks exploit yet unknown structures and representations, and that understanding their mode of operation may shed new light on the underlying mathematical problems.

On the other hand, applications of Reinforcement Learning (RL) and Large Language Models (LLMs) to automated theorem proving have also made drastic progress in the last six years [24, **?**, 22, **?**, 7, 32] The tandem workshop *Machine Learning and AI for Mathematics*, organized by François Charton (Paris), Jan de Gier (Melbourne), Amaury Hayat (Paris), Julia Kempe (New-York), Geordie Williamson (Sydney), aimed at discussing how AI methods can help mathematician advance mathematics. It was well attended, with over 40 participants. It focused on three angles:

- Using AI techniques to help mathematicians make advances on hard problems;
- Using mathematics to understand AI predictions;
- Using deep-learning models for automated theorem proving.

Since neural networks are notoriously good at spotting and learning weak signals and hidden structures, even in complex problems, they can be trained to suggest solutions to hard problems or counterexamples to conjectures, in the manner of an "artificial intuition". In this workshop we focused on several related lines of research. One considers problem solving as a translation task: the model is trained from pairs of problems and solutions to translate problems, encoded as a "sentence" (i.e. a sequence of symbols) into their solutions (also encoded as sequences of symbols), just as they would learn to translate a sentence from one language to another. A second line of research leverages the ability of machine learning techniques to learn efficient representations of large amounts of data, that reveal unexpected regularities, or relations between different quantities, that mathematicians then seek to understand. This was applied successfully, for instance, to knot theory [17, 21, 9] and representation theory [9]. A third line of research consists in using the ability of machine learning training procedures either to provide very efficient optimization in non-convex and/or high-dimensional frameworks or find interesting constructions in highly combinatorial spaces [6, 13]. Concerning automated theorem proving, formal proof assistants such as *Coq*, *Isabelle*, and more recently *Lean* [23], were developed to express mathematics with full logical precision and verify proofs [14, 4]. Although their rigorous syntax remains far from informal mathematical writing and is often tedious, a growing community has

nevertheless formalized large parts of both classical and modern mathematics, in particular in *Lean* [11, 10, 16, 8, 30, 5]. Recent advances in deep neural networks, especially LLMs, have renewed hopes that AI could assist in proof construction. Such tools could

(i) make formal languages more accessible by automatically completing routine steps;
(ii) translate informal human proofs into fully formal ones, ensuring correctness and consistency;
(iii) ultimately help mathematicians discover new proofs.

This workshop was a unique blend of researchers from all around the world with various backgrounds in Number Theory, Algebraic Geometry, Representation Theory, Dynamical Systems, Control Theory, Logic, Theoretical Physics, Knot Theory, Analysis of PDEs, Optimization, Formalized Mathematics, and, of course, Machine Learning.

Acknowledgement: The MFO and the workshop organizers would like to thank the National Science Foundation for supporting the participation of junior researchers in the workshop by the grant DMS-2230648, "US Junior Oberwolfach Fellows".

References

[1] Alberto Alfarano, François Charton, and Amaury Hayat. Global lyapunov functions: a long-standing open problem in mathematics, with symbolic transformers. Preprint, 2024.
[2] Kala Agbo Bidi, Jean-Michel Coron, Amaury Hayat, and Nathan Lichtlé. Reinforcement learning in control theory: A new approach to mathematical problem solving. In *3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*, 2023.
[3] 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.
[4] Kevin Buzzard. Proving theorems with computers. *Notices of the American Mathematical Society*, 67(11):1, 2020.
[5] Kevin Buzzard, Johan Commelin, and Patrick Massot. Formalising perfectoid spaces. In *P*roceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, pages 299–312, 2020.
[6] 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.
[7] Luoxin Chen, Jinming Gu, Liankai Huang, Wenhao Huang, Zhicheng Jiang, Allan Jie, Xiaoran Jin, Xing Jin, Chenggang Li, Kaijing Ma, et al. Seed-prover: Deep and broad reasoning for automated theorem proving. *arXiv preprint arXiv:2507.23726*, 2025.
[8] Johan Commelin and Robert Y Lewis. Formalizing the ring of witt vectors. In *P*roceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs, pages 264–277, 2021.
[9] 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.
[10] Floris van Doorn, Gabriel Ebner, and Robert Y Lewis. Maintaining a library of formal mathematics. In *International Conference on Intelligent Computer Mathematics*, pages 251–267. Springer, 2020.

[11] Floris van Doorn, Jakob von Raumer, and Ulrik Buchholtz. Homotopy type theory in lean. In *International Conference on Interactive Theorem Proving*, pages 479–495. Springer, 2017.
[12] Michael R Douglas. Machine learning as a tool in theoretical science. *Nature Reviews Physics*, 4(3):145–146, 2022.
[13] Bogdan Georgiev, Javier Gómez-Serrano, Terence Tao, and Adam Zsolt Wagner. Mathematical exploration and discovery at scale. *arXiv preprint arXiv:2511.02864*, 2025.
[14] P. C. Gilmore. A proof method for quantification theory: Its justification and realization. *IBM J. Res. Dev.*, 4(1):28–35, jan 1960.
[15] Sergei Gukov, James Halverson, and Fabian Ruehle. Rigor with machine learning from field theory to the poincaré conjecture. *Nature Reviews Physics*, pages 1–10, 2024.
[16] Alena Gusakov, Bhavik Mehta, and Kyle A Miller. Formalizing hall's marriage theorem in lean. *arXiv preprint arXiv:2101.00127*, 2021.
[17] Vishnu Jejjala, Arjun Kar, and Onkar Parrikar. Deep learning the hyperbolic volume of a knot. *Physics Letters B*, 799:135033, 2019.
[18] Karl Kunisch, Donato Vásquez-Varas, and Daniel Walter. Learning optimal feedback operators and their sparse polynomial approximations. *Journal of Machine Learning Research*, 24(301):1–38, 2023.
[19] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *International Conference on Learning Representations*, 2018.
[20] Guillaume Lample, Marie-Anne Lachaux, Thibaut Lavril, Xavier Martinet, Amaury Hayat, Gabriel Ebner, Aurélien Rodriguez, and Timothée Lacroix. Hypertree proof search for neural theorem proving. *Advances in neural information processing systems*, 2022.
[21] Jesse SF Levitt, Mustafa Hajij, and Radmila Sazdanovic. Big data approaches to knot theory: understanding the structure of the jones polynomial. *Journal of Knot Theory and Its Ramifications*, 31(13):2250095, 2022.
[22] 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. *CoRR*, 2025.
[23] Leonardo de Moura, Soonho Kong, Jeremy Avigad, Floris van Doorn, and Jakob von Raumer. The lean theorem prover (system description). In *International Conference on Automated Deduction*, pages 378–388. Springer, 2015.
[24] Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.
[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
[26] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.
[27] 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. *CoRR*, 2025.
[28] Yongji Wang, C-Y Lai, Javier Gómez-Serrano, and Tristan Buckmaster. Asymptotic self-similar blow-up profile for three-dimensional axisymmetric euler equations using neural networks. *Physical Review Letters*, 130(24):244002, 2023.
[29] Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021.
[30] Eric Wieser and Utensil Song. Formalizing geometric algebra in lean. *Advances in Applied Clifford Algebras*, 32(3):1–26, 2022.
[31] Geordie Williamson. Is deep learning a useful tool for the pure mathematician? *Bull. Amer. Math. Soc. (N.S.)*, 61(2):271–286, 2024.

[32] Huajian Xin, ZZ Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, et al. Deepseek-prover-v1. 5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. In *The Thirteenth International Conference on Learning Representations.*

Workshop: MATRIX-MFO Tandem Workshop: Machine Learning and AI for Mathematics

Table of Contents

Abstracts

A Mathematician's AI Wishlist

Melanie Matchett Wood

"I want AI to do my laundry and dishes so that I can do art and writing, not for AI to do my art and writing so that I can do my laundry and dishes." – Joanna Maciejewska

Question: What can we imagine AI doing for mathematicians that would make us more effective and efficient at doing high-quality mathematics?

1. Math from an LLM at graduate/research level is disturbingly unreliable

- Largely correct on calculus, linear algebra. Risk for young mathematicians to trust too much.
- Undergraduate level group theory question: Chat GPT 5 gave me both a proof and a counterexample, Gemini 2.5 Pro gave a correct proof
- Chat GPT 5 Pro, Gemini 2.5 Pro repeatedly gives false statements, cosmetically similar to true statements, in graduate level topics of group cohomology, cohomology of schemes.
- Will it "learn more math" or will low coverage always be an issue?
- How do we verify?

2. The Wishlist

Some basic features of a math assistant LLM, all of which are lacking in current reasoning LLM models.

(1) Find a reference for a statement, without hallucination.
(2) Find where a notation is defined in a reference
(3) Find mathematical mistakes
(4) An LLM that speaks like a mathematician (reflecting the need to have proofs and specific references in mathematics, understanding a result and counterexample can't both be true, understanding that proving something stronger proves something weaker, trained to be skeptical, trained to find mistakes, distinguishing conjectures and theorems)

Given the amount of content LLMs produce, autoformalization will be necessary to make sure mathematics does not become awash is a sea of nonsense and unproven assertions. One possible low-hanging fruit would be natural language computations and auto-formalization. This would allow for exploring examples in more abstract fields. An agent that could do this would list good examples to try to computations on, be able to do in natural language the kind of computation a graduate student might spend a few days on–procedures for which there is no articulated algorithm, but are somewhat straightforward perhaps with a few adhoc tricks. Then these natural language computations would be natural candidates to formalize into Lean, as computations are often more accessible to formalize.

We will eventually need autoformalization more generally.

(1) Check my arguments: I write a natural language proof. Can AI confirm the details in a Lean verified proof?
(2) Check LLM's arguments: Don't trust, and verify.
(3) Check all the arguments: I think the future is going to be filled with highly persuasive, incorrect mathematical text, and formalization is going to be required to keep mathematics mathematics.

Boosting sphere packings using diffusion and flow-matching models

Gergely Bérczi

(joint work with Adam Zsolt Wagner, Jonas Klüver, Baran Hashemi)

Aim. We explore whether modern generative models can *construct* high-quality finite sphere packings in a bounded domain, and thereby suggest patterns that may inform densest packings and lower bounds in higher dimension. The talk reported initial progress on a hybrid pipeline that alternates *local optimization* with *global generation* from diffusion/flow-matching models, in the spirit of a "local → global → local" boosting loop.

Context. For dimensions 8 and 24, Viazovska and Cohn–Kumar–Miller–Radchenko–Viazovska proved optimality of the E_8 and Leech lattice packings, respectively; in general dimension, recent work gives improved (asymptotic) lower bounds and new lattice constructions.[1] While these landmarks concern the infinite-volume density, we focus on the practical surrogate of packing congruent balls in a unit cube, which avoids boundary-volume estimation issues and yields directly checkable candidates.

Pipeline ("DiffuseBoost").

- *Local search (Physics-Push / SRP):* Starting from random centers in $[r, 1-r]^d$, we iteratively repel overlaps (force-based "physics push"), or minimize an *overlap energy* by stochastic repulsion (SRP) with annealed, normalized gradient steps followed by a box-constrained L-BFGS-B polish. This produces large sets of *near-valid* packings that concentrate near good radii and provide diverse training data.
- *Global generation (conditional flow matching):* We train a time-dependent vector field $u_\theta(x, t)$ that transports easy seeds (e.g. jittered lattice samples) to the empirical distribution of good packings along a simple probability path. Training is "simulation-free" regression to an analytic conditional velocity; sampling integrates the learned ODE from $t = 0$ to $t = 1$.

[1] See, e.g., M. Viazovska: The sphere packing problem in dimension 8 (Ann. Math. 185 (2017)) and Cohn et al.: The sphere packing problem in dimension 24 (Ann. Math. 185 (2017) for $d = 8, 24$; Campos–Jenssen–Michelen–Sahasrabudhe (arXiv:2312.10026) and Klartag (arXiv:2504.05042) for high-dimensional lower bounds. These references are for context; our contribution is algorithmic and empirical.

- *Projection-corrected sampling (PCFM):* During sampling we interleave *predict* → *project* → *correct*. Each ODE step is projected onto the non-overlap constraints (pairwise and wall), followed by a small proximal relaxation toward the learned flow to avoid fighting the projection.
- *Final refinement:* A short physics-push/SRP stage removes residual contacts and aligns candidates with nearby local optima.

Preliminary observations. Across moderate numbers of balls (e.g. a few dozen in $d = 3$ and hundreds of circles in $d = 2$), the diffusion/flow stage *shifts* the distribution of minimal pairwise separations to the right, compared with local search alone, and increases the yield of near-record configurations. Iterating the loop further boosts the right tail ("boost effect"). Architecturally, flow-matching models are more sensitive to training choices than transformer encoders, but give better global moves once tuned. (Plots and examples appeared in the talk.)

Relation to PatternBoost. Our strategy adapts the "PatternBoost"[2] idea — local search to seed a small generative model, then local repair—which we previously used to slightly improve bounds for bootstrap percolation on hypercubes. Here, replacing sequence models by diffusion/flow significantly improves coverage of the design space and the quality of de novo proposals.

Open questions and conjectures.

(1) *Generalization across n and d.* Can a single conditional model, trained on a mixture of ball counts and dimensions, extrapolate to larger instances while maintaining or improving minimal separations after projection?
(2) *From finite boxes to periodic packings.* Training on periodic cells with lattice variables and hard contact constraints: will the pipeline discover known extremal lattices in $d = 4, 5, 6, 7$ or propose plausible non-lattice candidates for $d \geq 9$?
(3) *Improve PCFM.* Prove convergence guarantees for projection-corrected flow matching under non-convex, inequality constraints of contact type; relate to score-based samplers with constraint manifolds.

Outlook. Even when not producing ultimate records, the sampler supplies high-quality, diverse seeds that substantially reduce time-to-solution for downstream local optimization. We view this as a *construction heuristic* that complements analytic bounds and may reveal structural patterns worth formalizing.

Acknowledgements. I thank the organizers and participants for a stimulating workshop, and my collaborators Adam Zsolt Wagner, Jonas Klüver and Baran Hashemi for many contributions.

[2]François Charton, Jordan S. Ellenberg, Adam Zsolt Wagner, Geordie Williamson: PatternBoost: Constructions in Mathematics with a Little Help from AI, arXiv:2411.00566

An Introduction to ICARM

Matthew R. Ballard

The Institute for Computer-Aided Reasoning in Mathematics (ICARM) is a new NSF Mathematical Sciences Research Institute designed to accelerate mathematical research by leveraging artificial intelligence and computer-aided reasoning. Located at Carnegie Mellon University and operating as a 3-year pilot program, ICARM addresses the rapidly evolving landscape of AI technologies in mathematics through a distinctive mission centered on three pillars: empowering mathematicians to keep mathematics central in AI-driven discovery, fostering collaboration across disciplines and career stages, and ensuring equitable access to emerging technologies.

A key innovation of ICARM is the creation of Innovation Engineers—a new professional role bridging mathematics and technology. These technical staff work alongside mathematicians to provide training, documentation, and tutorials; offer hands-on support through office hours, site visits, and online platforms; maintain communal resources including formal libraries like Mathlib, benchmark datasets, and computational infrastructure; develop and disseminate tools spanning formalization, automated reasoning, and machine learning for mathematics; and collaborate on longer-term research projects.

ICARM's scientific activities encompass multiple formats designed to engage the broad mathematical community. Two-week summer schools offer hands-on training in formalization, automated reasoning, and machine learning for mathematics, with the first week dedicated to tutorials and guided exercises and the second week focused on collaborative projects. These schools welcome participants from PhD students to senior mathematicians, with travel, housing, and stipends available to support broad participation. One-week research workshops bring together 20–30 participants for highly interactive sessions that minimize traditional lectures in favor of collaborative work, aiming to spark new projects and long-term partnerships. Research group visits support short- to medium-term stays (2 weeks to 1 semester) for individuals or small groups who require tools, training, or collaborations not available at their home institutions, with Innovation Engineers providing expert guidance throughout.

The institute benefits from substantial infrastructure support, including computing resources from Carnegie Mellon University and the Pittsburgh Supercomputing Center, commitments from industry partners, and crucially, the expertise and engagement of the mathematical community itself. ICARM actively invites mathematicians to participate in summer schools and workshops, propose topics for future programs, apply for collaborative research visits, work directly with Innovation Engineers, and help shape the institute's direction. Through these mechanisms, ICARM aims to build a national and international hub for AI and computer-aided reasoning in mathematics, bridging mathematicians, computer scientists, academia, industry, education, and outreach—ensuring that as AI transforms mathematical practice, mathematicians remain at the center of mathematical discovery.

AI for Mathematics – From Digitization to Intelligentization

Bin Dong

The Bottlenecks in Mathematical Research

In my talk, I wanted to begin by arguing that modern mathematical research, for all its successes, faces several significant bottlenecks that limit our efficiency. One of the most critical is the process of proving and verifying results. Proofreading mathematical papers is a tedious and error-prone task. Even for the most accomplished mathematicians, small errors can creep into proofs written in natural language, sometimes going undiscovered for years. As was discussed in a blog post by Terence Tao, there is hope that AI tools may soon help automate parts of this process, but we are not there yet.

Furthermore, we often say that research is done by "standing on the shoulders of giants," but it is not always easy to locate these shoulders. A quick search on the internet or asking experts can be helpful, but it is difficult to be certain whether a theorem or an idea is genuinely new. Finally, tackling a new problem often requires learning entirely new mathematical concepts and tools, which can be a time-consuming and challenging endeavor. These are precisely the areas where I believe AI has the potential to help us achieve breakthroughs.

From AI-Assisted Discovery to a Mathematical "Apprentice"

Early explorations into "AI for Mathematics" have shown great promise. A well-known study by DeepMind and Geordie Williamson [1] demonstrated how machine learning can augment a mathematician's intuition to discover new patterns and conjectures. Inspired by this, my collaborator Xu-Hua He and I used similar methods to explore the dimension formula of Affine Deligne-Lusztig varieties (ADLV) [2]. While we did not solve the problem completely, the machine learning insights led us to prove a new theorem establishing an upper bound for the error between the virtual and true dimensions.

This approach of building specialized AI tools for specific problems is effective, but it does not address the core bottleneck of theorem proving and verification, which remains a human task. A more systematic solution, I believe, lies in leveraging large language models (LLMs). The goal here is more ambitious: to train an "AI apprentice." However, for an LLM to be genuinely useful for research-level mathematics, it must overcome its current limitations, such as unreliability and the challenge of verification; while verifying proofs for lower-level mathematics is feasible, this task becomes increasingly difficult for the advanced topics.

The success of LLMs is built upon the concept of foundation models, which are designed to generalize across a wide variety of tasks. The key technology enabling this is the Transformer architecture. Transformers manage to handle diverse, often multimodal, data and tasks through a few core principles: tokenization unifies different modalities into a common representation, the objective of next-token prediction provides a universal framework for different tasks, and the architecture

itself scales remarkably well with more data and computational resources. This has proven to be incredibly effective for natural language, and there is good reason to believe it will be for mathematics as well, because, at its core, mathematics is also a language.

The Key: Digitalization of Mathematics with Lean

The key to unlocking the potential of AI for mathematics is, in our view, the further digitalization of mathematics. By translating mathematical knowledge from natural language into a formal language that a computer can understand and verify, we can build a "mathematical reasoning simulator." This simulator would provide the ideal environment to train AI models to first mimic, and hopefully one day surpass, human reasoning.

Currently, one of the most mature formal language systems is Lean, a proof assistant developed by Microsoft. Its main mathematics library, `mathlib`, is a massive, collaborative effort containing over 190,000 theorems and 90,000 definitions. However, working with Lean presents a steep learning curve. This motivated our AI4M team at Peking University (a diverse group of pure mathematicians, applied mathematicians, and engineers) to develop tools that bridge the gap between informal natural language mathematics and its formal counterpart.

Recent Progress from the AI4M Team at PKU

Our team is focused on a long-term goal of resolving open problems in mathematics, starting with algebra where `mathlib` is most mature. To guide our progress, we have developed a series of formal benchmark suites of increasing difficulty: FATE-M (entry level), FATE-H (Master's level), and FATE-X (close to research level) [3]. Along the way, we have produced several tools and resources that we believe are of interest to the community.

- **LeanSearch**: An effective tool to help users find relevant theorems and definitions within the vast `mathlib` library, addressing the common problem of spending too much time just searching for the right lemma [4].
- **Herald**: A high-quality dataset of paired natural language and formal language statements, created to train translation models for autoformalization [5]. Our Herald Translator model significantly outperforms existing open-source models on graduate-level mathematics.
- **REAL-Prover**: An open-source, step-by-step theorem prover for Lean 4 [7]. On the ProofNet dataset and our new `FATE-M` benchmark, it has surpassed current state-of-the-art performance. We have also released it as a lightweight tactic in Lean, `reap`, to help users draft formal proofs interactively [6].

These tools represent our initial steps toward creating a unified "copilot" for interactive theorem proving.

A Vision and Open Questions for the Future

Looking ahead, our vision for AI in mathematics extends beyond proof assistance. Some of the most valued achievements in the field involve building bridges, creating meaningful connections between different areas of mathematics. However, it's extremely difficult for any single mathematician or team to master all these areas; it is simply beyond human limits. This is a barrier AI can help us overcome. Our hope is that AI will be able to effectively integrate the vast knowledge of mathematics and become a powerful research assistant, helping to uncover the deep connections that drive mathematical progress.

As William Thurston wrote in his classic article, "On Proof and Progress in Mathematics" [8], the goal of mathematics is not just proof, but understanding. As AI matures to handle the tedious and mechanical aspects of proof verification, it will free us to focus on the more creative and enjoyable activities: building intuition, asking deep questions, and gaining true understanding.

In this spirit, I would like to propose two key questions to promote new research:

(1) How can we build an agentic AI system that effectively mimics the full workflow of a human mathematician¡[a]from exploring examples and formulating conjectures to strategically proving theorems¡[a]and can it autonomously solve problems from research-level benchmarks like FATE-X?
(2) What is the most effective architecture for combining the pattern-recognition strengths of neural networks with the rigorous, logical deduction of symbolic reasoners to produce novel, human-readable proofs?

Our work is still in its early stages, and the results are provisional. However, we are optimistic about the future and welcome collaborations.

References

[1] A. Davies et al., *Advancing mathematics by guiding human intuition with AI*, Nature **600** (2021), 70–74.
[2] B. Dong, X. He, P. Jin, F. Schremmer, and Q. Yu, *Machine learning assisted exploration for affine Deligne-Lusztig varieties*, Peking Mathematical Journal (2024), 1–50.
[3] The AI4M Team, *FATE: A Formal Algebra Benchmark Suite*, Blog post on "FrenzyMath", https://frenzymath.com/blog/fate/, 2025.
[4] G. Gao, H. Ju, J. Jiang, Z. Qin, and B. Dong, *A semantic search engine for mathlib4*, in: Proceedings of the 2024 Conference on Empirical Methods for Natural Language Processing (EMNLP 2024), arXiv:2403.13310.
[5] G. Gao, Y. Wang, J. Jiang, Q. Gao, Z. Qin, T. Xu, and B. Dong, *Herald: A Natural Language Annotated Lean 4 Dataset*, in: Proceedings of the Thirteenth International Conference on Learning Representations (ICLR 2025), arXiv:2410.10878.
[6] The AI4M Team, *reap: A lightweighted Lean tactic for drafting formal proofs*, Blog post on "FrenzyMath", https://frenzymath.com/blog/reap/, 2025.
[7] Z. Shen et al., *REAL-Prover: Retrieval Augmented Lean Prover for Mathematical Reasoning*, arXiv preprint arXiv:2505.20613, 2025.
[8] W. P. Thurston, *On proof and progress in mathematics*, Bulletin of the American Mathematical Society **30** (1994), 161–177.

The Carleson project: a collaborative formalization

María Inés de Frutos-Fernández, Floris van Doorn
(joint work with Lars Becker, Leo Diedering, Sébastien Gouëzel, Asgar Jamneshan, Evgenia Karunus, Edward van de Meent, Pietro Monticone, Jasper Mulder-Sohn, Jim Portegies, Joris Roos, Michael Rothgang, Rajula Srivastava, James Sundstrom, Jeremy Tan, Christoph Thiele)

Trigonometric series represent functions as possibly infinite linear combinations of pure frequencies. They gained particular prominence through the work of Fourier, who used them in his analytical theory of heat, thereby establishing them as a tool for solving partial differential equations. Fourier also made the groundbreaking claim that a wide range of functions could be represented using trigonometric series. This sparked the interest of many mathematicians, including Dirichlet, who gave some rigorous conditions for convergence of Fourier series, as trigonometric series are now called. Dirichlet also opened a branch of analytic number theory partially inspired by the ideas of Fourier. Nowadays, Fourier analysis plays an important role in many areas of mathematics.

With Euler's formula to represent pure frequencies in mind, a trigonometric polynomial can be expressed as

$$S_N(x) := \sum_{n=-N}^{N} c_n e^{inx} \,. \tag{1}$$

The Fourier series is then defined as the limit f of such a sequence S_N as N tends to ∞. Fourier's vision to represent rather general functions raises two fundamental questions. The first question is to identify the appropriate choice of coefficients c_n to use to represent a given f. The second question addresses the convergence of S_N to f.

The first question has a fairly canonical and standard answer, provided by the Fourier integral formula:

$$c_n := \widehat{f}_n := \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-inx}\, dx, \tag{2}$$

where the precise interpretation of the integral depends on the chosen theory of integration. For a continuous function f, Riemann's notion of the integral suffices. More generally, if f is a distribution in the sense of Schwartz, supported in $[0, 2\pi]$ the integral can be understood as an evaluation against the periodic test function e^{-inx}. The more general definition reduces to the simpler one within the respective more restrictive domain. Hence, the Fourier coefficients given by (2) serve as a universal choice. This choice is unique in several respects, in particular if one is to exactly reproduce a trigonometric polynomial f in the form (1).

The second question of convergence bifurcates into the question of pointwise convergence of the series (1) (with coefficients given by (2)) for a given x on the one hand and convergence of the functions S_N to the function f in a suitable function space with corresponding topology on the other hand. There are at least

as many function spaces for the question of convergence as there are different definitions of the integral elaborated earlier. There are some very canonical answers to the convergence question in function spaces, albeit not known at the time of Fourier and Dirichlet. One example is convergence in the Hilbert space sense for f in $L^2(0, 2\pi)$, as discovered in the first decade of the twentieth century as a consequence of the rapid development of Lebesgue integration theory. For some other natural spaces, such as $L^1(0, 2\pi)$, there is no guarantee of convergence in the norm of that space even if f is in the space.

In contrast to these examples of function spaces with a very natural theory of convergence of Fourier series in the topology of the function space, there are no similarly elegant solutions to the characterization of pointwise convergence. In particular, the space of functions f such that the sequence $S_N(x)$ converges for every x does not have a good characterization in terms of f itself. Similarly, the space of all functions f such that the sequence of coefficients $\widehat{f_n}$ is absolutely summable has also no good characterization.

When the Fourier integral is defined in the Lebesgue sense and $f \in L^1(0, 2\pi)$, then the function f itself is meaningful not everywhere but only pointwise almost everywhere in the Lebesgue sense. The question of pointwise convergence to f for all x then becomes meaningless, and instead one asks for almost everywhere convergence. Such convergence was conjectured by N. Luzin for the space $L^2(0, 2\pi)$, and proven by Carleson in the 1960s [2]. In particular, Carleson also proved the more elementary statement

Theorem 1 (classical Carleson)**.** *Let f be a 2π-periodic complex-valued continuous function on $\mathbb{R}$. Then for almost all $x \in \mathbb{R}$ we have*

$$\lim_{N\to\infty} S_N f(x) = f(x), \tag{3}$$

where $S_N f$ is the N-th partial Fourier sum of f defined in (1) with coefficients (2).

Here, almost every x means in the Lebesgue sense, i.e., for every $\epsilon > 0$ the set of x where convergence fails can be covered by a sequence of intervals such that the sum of the lengths of these intervals is less than ϵ. While Carleson had proven the more general Luzin conjecture for functions in $L^2[0, 2\pi]$, even the more elementary statement for continuous functions was not known before Carleson's work. Moreover, until now, the elementary statement has not seen any substantially easier proof than those generalizing to L^2, partially because there is no readily usable criterion on the level of Fourier coefficients to distinguish between continuous functions and L^2 functions.

In the 1970s, Fefferman gave an alternative proof of Carleson's theorem via an a priori bound for Carleson's operator, the maximally modulated singular integral

$$Tf(x) := \sup_N \int_{-\pi}^{\pi} e^{iNy} f(x-y)\frac{1}{y}dy\,. \tag{4}$$

Various strengthenings of Fefferman's estimates for Carleson's operator have appeared since.

In this talk, we describe a formalization in Lean of a further generalization of the polynomial Carleson operator towards doubling metric measure spaces, which is a new theorem described in the preprint [1]. This new result is an axiomatic approach to Carleson type theorems on doubling metric measure spaces, which is suitable for formalization and provides a good route towards the classical theorem, which we also formalize as a corollary.

Early drafts of the preprint [1] existed in summer 2023. Based on this, a first draft of a blueprint for the formalization was written in the first half of 2024, containing a much more detailed proof, which involved increasing the size by a factor of four, and adding the derivation of Carleson's classical result. In June 2024, Floris van Doorn launched a public website to post the blueprint, using the open-source software `leanblueprint` developed by Patrick Massot, calling for contributions to formalize the proof. The goal was to formalize the blueprint in the Lean proof assistant [3], building on top of its mathematical library `Mathlib` [4]. The work was split up into about 180 tasks, to be claimed by individual contributors. Most tasks were to formalize the proof of a single lemma from the blueprint, and some were to develop basic theory or refactor existing code. The contributors adapted the blueprint to fix some gaps found during the formalization and gave feedback that led to discussions about the proof. This even resulted in a few changes to the general setup and the main theorems. All of the gaps found required only fairly localized changes to the blueprint, indicating that the initial blueprint was already of high quality. The formalization was completed in July 2025, and the latest version can be found on Github.

Everyone that completed a substantial amount of tasks is included as a coauthor of the blueprint. The authors acknowledge contributions in the form of small formalization additions, pointing out corrections to the blueprint, or supplying ideas to the Lean efforts by the following people: Michel Alexis, Bolton Bailey, Julian Berman, Joachim Breitner, Martin Dvořák, Georges Gonthier, Aaron Hill, Austin Letson, Bhavik Mehta, Eric Paul, Clara Torres, Dennis Tsar, Andrew Yang, Ruben van de Velde.

References

[1] L. Becker, F. van Doorn, A. Jamneshan, R. Srivastava, and C. Thiele., *Carleson operators on doubling metric measure spaces*, arXiv preprint 2405.06423v2 (2024).

[2] L. Carleson. *On convergence and growth of partial sums of Fourier series*, Acta Mathematica, **116** (1966), 135–157.

[3] L. de Moura and S. Ullrich., *The Lean 4 Theorem Prover and Pro gramming Language*, Lecture Notes in Computer Science **12699** (2021), 625–635.

[4] The mathlib Community, *The Lean mathematical library*, In Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (2020), 367–381.

Finding interesting mathematical objects with ML tools

Adam Zsolt Wagner

Proofs are at the heart of mathematics, they are certainly one of the most important things mathematicians care about. But there are many situations where the focus is not actually on proofs, but rather on how to find interesting mathematical constructions instead. This happens for example when we encounter a conjecture that we believe to be false. In this case, our task is to come up with a weird graph, matrix, or set system, the demonstrates that the conjecture is false. Usually once we have found a counterexample it is easy to verify it actually works, the hard bit is how to find this construction in the first place. Another similar situation happens when we face a problem where we don't even know what the optimal constructions might look like. In these cases it would be extremely helpful if we could ask an oracle to tell us the optimal constructions for, say, $n = 10, 20, 50$, or 100. Once we can see these constructions, we can stare at them, start spotting patterns, which will eventually lead to new observations, new conjectures, new theorems, and new mathematics.

Gil Kalai once said "The methods for coming up with useful examples in mathematics [...] are even less clear than the methods for proving mathematical statements". Given this quote, and how important I perceive this theme of finding constructions in mathematics to be, I firmly believe that if we can get simple and useful computational tools into the hands of mathematicians, that will lead to lots of new mathematics in the future.

This talk is focused on tools that in some way use machine learning. My goal is to present a spectrum of different tools mathematicians can use in their research, then compare them, so that at the end of the talk we have a better understanding of what tools would be best for the problems we care about. I start with the simplest possible learning-based tools, and work my way up to more modern LLM-based methods such as AlphaEvolve.

The very first method I present is simple, vanilla reinforcement learning. I demonstrate that while this method is extremely simple, in some cases it is good enough and can lead to discoveries of new objects without much effort. Often this method is not good enough to find a counterexample or a good construction all by itself, but it might still give us insights that we can use. A mathematicians could then take these new insights and find a counterexample by themselves. This human-AI collaboration is often what leads to the best results, with every method we discuss here.

The second method I present is PatternBoost. When we humans think about a program we don't just think about it locally. We might create a global picture of the optimal constructions in our head, then zoom in to check if all the local restrictions are satisfied, if not we would zoom out and revise our global plan, and so on. We can do a little bit of this local-global search by modifying our previous setups slightly, and I will show that this leads to much better results.

Finally, in the second half of the talk, I discuss language based methods, specifically FunSearch and AlphaEvolve. The idea is the following. We understand what

the basic idea of a local search in the space of graphs looks like: we have a graph, we compute its score, then we add/remove an edge, compute the score of the new graph, and if the score increased then we replace the old graph with the new one and repeat. In other words, we follow the path where the score goes up, always jumping to nearby graph.

We will now do the same thing in language space: instead of working with graphs, we will work now with python programs that generate graphs. The score of a python program is the score of the graph it produces. We can do the same local search as before: jumping from one program to a nearby program (where "nearby" is up to an LLM to judge), and following a path where the score goes up. This has advantages and disadvantages. For many natural problems, the optimal construction has a simple description in python, whereas the many local maxima that standard local search methods might get stuck in, do not. Moreover, even if there is only one unique optimal construction, there are often many natural python programs representing it. So while we might still be looking for a needle in a haystack, with this method the needle could be bigger, and there could be many more of them in the haystack. The downside is that for every new construction we have to make an LLM call, which is often slower and more expensive than standard local search ran on a CPU.

A solution to this, by AlphaEvolve, is to combine all methods we have mentioned so far. Instead of walking around in the space of python programs that generate constructions, we will walk around in the space of search functions, that have a fixed time budget to explore many graphs in any way they want, and only after the time is up do they have to return the best construction they have found. This solves the disadvantage mentioned above: a single LLM call generates a search method that can unleash a flurry of cheap exploration. AlphaEvolve will then explore the space of heuristic search algorithms to find one that works well for the problem.

There are several additional ideas one can add onto this setup, which we discuss. But one theme I emphasize is the importance of the expert guidance. In the prompt given to AlphaEvolve, we can always put some hints about what the optimal constructions might look like. A better hint yields a better result, and I show some experiments highlighting just how important a good initial advice is. I will also show that AlphaEvolve always retains the melody of the original advice it was given, it simply tries to squeeze as much out of it as possible. So this tool can be used by mathematicians to test big picture ideas they might have on a problem, without having to work out all the details by themselves.

One thing that is clear to me is that there are lots of capabilities of these tools that are not yet fully explored and there are lots of low hanging fruits. I am very excited to see what you will all come up with in the next years!

References

[1] Charton, F., Ellenberg, J. S., Wagner, A. Z., & Williamson, G. (2024). Patternboost: Constructions in mathematics with a little help from ai. arXiv preprint arXiv:2411.00566.

[2] Novikov, A., Vũ, N., Eisenberger, M., Dupont, E., Huang, P.S., Wagner, A.Z., Shirobokov, S., Kozlovskii, B., Ruiz, F.J., Mehrabian, A. and Kumar, M.P., 2025. AlphaEvolve: A coding agent for scientific and algorithmic discovery. arXiv preprint arXiv:2506.13131.
[3] Romera-Paredes, B., Barekatain, M., Novikov, A., Balog, M., Kumar, M.P., Dupont, E., Ruiz, F.J., Ellenberg, J.S., Wang, P., Fawzi, O. and Kohli, P., 2024. Mathematical discoveries from program search with large language models. Nature, 625(7995), pp.468-475.
[4] Wagner, Adam Zsolt. "Constructions in combinatorics via neural networks." arXiv preprint arXiv:2104.14516 (2021).

Reasoning machines with RL - Experiences from the field

Albert Jiang

In this talk, we introduce Magistral, Mistral's first reasoning model, which leverages a scalable reinforcement learning (RL) pipeline. Unlike previous approaches that rely on distilled reasoning traces from prior models, Magistral is trained from scratch using a ground-up approach, leveraging Mistral's own models and infrastructure. We demonstrate the effectiveness of pure RL training for large language models (LLMs), present a method to enforce multilingual reasoning, and show that RL on text data alone can maintain and even improve the initial model's capabilities.

Our training methodology involves using Group Relative Policy Optimization (GRPO) with several modifications to enhance stability and exploration. These modifications include eliminating the KL divergence penalty, normalizing the loss, and relaxing the trust region's upper bound. The reward shaping strategy ensures the model adheres to proper format, length, and language usage.

The infrastructure for distributed RL training coordinates trainers, generators, and verifiers. Trainers maintain the model weights and perform gradient updates, generators perform roll-outs, and verifiers evaluate completions. The system prioritizes efficiency and operates generators continuously at maximum throughput without waiting for trainers.

Our data curation process involves extensive filtering and difficulty assessment for both math and code problems. For math, problems are filtered based on format and difficulty. For code, problems are selected based on the availability of solutions and tests.

Experimental results show significant performance improvements on reasoning benchmarks, including a nearly 50% boost in AIME-24 (pass@1) for Magistral Medium. Magistral Small, trained with RL on top of reasoning SFT bootstrapping, also demonstrates substantial performance gains. The model's multilingual capabilities are evaluated on translations of the AIME 2024 benchmark.

Ablation studies provide insights into the effects of different training parameters and choices, such as batch and minibatch sizes, and advantage normalization methods. These studies show that performance is not strongly dependent on batch size but degrades when there are more than two minibatches in a batch.

Our analysis investigates the dynamics of RL training and the impact on multimodal capabilities. The findings suggest that RL training improves multimodal reasoning and maintains tool-calling and instruction-following capabilities.

We conclude with a discussion of future research directions, including scaling up RL training, optimizing loss functions, and exploring new applications in tool-use, multimodality, and agents.

CayleyPy – Python AI-based library for google size Cayley graphs

Alexander Chervov

(joint work with A. Soibelman, S.Galkin, D. Fedoriaka, E. Konstantinova, A. Naumov, I. Kiselev, A. Sheveleva, I. Koltsov, S. Lytkin, A. Smolensky, F. Levkovich-Maslyuk, R. Grimov, D. Volovich, H. Isambert *et al.*)

We present "CayleyPy" project applying artificial intelligence methods to problems in group theory. We announce the first public release of CayleyPy, an open-source Python library for computations with Cayley and Schreier (coset) graphs. Compared with state-of-the-art systems based on classical methods, such as GAP and Sage, CayleyPy handles significantly larger graphs and performs several orders of magnitude faster.

Using CayleyPy we obtained about 200 new mathematical conjectures on Cayley and Schreier graphs, with special regard to their diameters and growths.

For many Cayley graphs of symmetric groups S_n we observe quasi-polynomial diameter formulas: a small set of quadratic or linear polynomials indexed by n mod s, and conjecture that it is a general phenomenon. These lead to efficient diameter computation, despite the problem being NP-hard in general. We propose refinement of the Babai-type conjecture on diameters of S_n: $\frac{1}{2}n^2+4n$ upper bounds for the diameters in the standard undirected case, as compared to prior conjectural bounds of $O(n^2)$. We also provide explicit generator families, related to involutions in a simple "square-with-whiskers" pattern, which we conjecture to maximize the diameter; extensive (and in some cases exhaustive) search confirms this for all $n \leq 15$. We conjecture an answer to the celebrated open question raised by the "founding father of Soviet cybernetics" V. M. Glushkov in 1968: the diameter of the directed Cayley graph generated by the left cyclic shift and the transposition of the first two elements is equal to $(3n^2-8n+9)/4$ for n odd, and to $(3n^2-8n+12)/4$ for n even.

For nilpotent groups we conjecture an improvement of J. S. Ellenberg's results on the diameters of the upper unitriangular matrices over Z/pZ, presenting a phenomenon of linear dependence of the diameter on p. Moreover, the growth for nilpotent groups is conjectured to closely follow Gaussian distributions, that is, to exhibit a central limit phenomenon similar to the results of P. Diaconis for S_n.

Some of our conjectures are "LLM-friendly" — they can be stated as sorting problems, which are easy to formulate for LLM, and their solutions can be given by an algorithm or by a Python code, which is easy to verify, so they can be used

to test LLM's abilities to solve research problems. To benchmark various methods of path-finding on Cayley graphs we create more than 10 benchmark datasets in the form of Kaggle challenges, making benchmarking easy and public to the community. CayleyPy works with arbitrary permutation or matrix groups, and supports a pre-defined collection of more than a hundred generators including puzzle groups. Our code for direct growth computation outperforms similar functions on the standard computer algebra system GAP/SAGE up to 1000 times both in speed and in maximum sizes of the graphs that it can handle.

References

[1] A. Chervov, *et al. CayleyPy Growth: Efficient growth computations and hundreds of new conjectures on Cayley graphs (Brief version)*, https://arxiv.org/abs/2509.19162
[2] A. Chervov, *et al. CayleyPy RL: Pathfinding and Reinforcement Learning on Cayley Graphs*, https://arxiv.org/abs/2502.18663
[3] A. Chervov, *et al. A Machine Learning Approach That Beats Large Rubik's Cubes*, https://arxiv.org/abs/2502.13266

Generalization in the symmetric group

Alexandr Garbali
(joint work with Max Petschack and Jan de Gier)

We study [1] the capacity of transformer neural networks to learn the structure of the symmetric group S_n when trained only on smaller subgroups $S_m \subset S_n$. The task is to map a word in generators of S_n (expressed as transpositions) to the corresponding permutation in one-line notation, without explicitly encoding group relations. This provides a natural and noise-free benchmark for testing generalization and interpretability in machine learning applied to abstract algebra.

Methodology. Training data consist of pairs (x, p), where $x = (x_1, \dots, x_N)$ is a word in generators of S_m and p is the resulting permutation of $(1, \dots, n)$. To allow comparison across group sizes, we embed words from S_m into S_n through *identity augmentation*: unreduced words are modified by insertions of segments of words representing the identity permutation so that all inputs have fixed context length $N = n(n-1)/2$. The transformer is then trained autoregressively to predict permutation tokens.

Two regimes are explored:

(1) general transpositions $s_{i,j}$, with training on S_{10} and testing on S_{25};
(2) adjacent transpositions $s_i = s_{i,i+1}$, with training on S_{10} and testing on S_{16}.

In both cases we used a *partitioned window* method to embed representations of S_m into representations of S_n to prevent the network from memorizing local substructures.

Results. In both regimes the models achieve near 100% accuracy when generalizing from S_m to S_n, indicating that structural features learned on smaller groups transfer effectively to larger ones. Analysis of the learned embeddings shows that

- token embeddings capture algebraic relationships between generators (for example, $s_{i,j}$ correlating with $s_{i,k}$ and $s_{j,i}$);
- positional embeddings display emergent correlation patterns reflecting the maximal reduced word length of the training subgroup S_m, revealing a learned length scale $L_m = m(m-1)/2$ in the covariance of position vectors.

Discussion. These results demonstrate that transformer models can internalize the algebraic structure of the symmetric group and generalize beyond the distribution seen in training. The embedding method provides a scalable strategy for symbolic problems with natural inclusion maps between vocabularies, and the positional covariance analysis offers a new probe into the model's inductive bias. Potential extensions include applications to Hecke algebras, Kazhdan–Lusztig polynomials and braid group word problems, where similar embedding hierarchies (may) exist.

References

[1] Max Petschack, Alexandr Garbali and Jan de Gier *Learning the symmetric group: large from small*, arXiv preprint arXiv:2502.12717 (2025).

Tropical Attentions: Novel Algorithmic Reasoning for Combinatorial Algorithms

Baran Hashemi

Dynamic programming algorithms for combinatorial optimization problems often involve taking max, min, and classical addition in their recursion algorithms. The associated value functions correspond to convex polyhedra in $(\max, +)$ semiring. Existing Neural Algorithmic Reasoning (NAR) models, however, rely on softmax-normalized dot-product reasoning core, where the smooth operator smooths everything, blurs these sharp polyhedral structures and collapses when evaluated in out-of-distribution (OOD) settings. We introduce Tropical Attention, a novel reasoning core that operates natively in the $(\max, +)$ semiring of Tropical geometry, with Tropical Hilbert projective metric as the distance measure. We prove that Tropical Attention can universally approximate tropical circuits of DP-type combinatorial algorithms. Our results demonstrates that Tropical Attention restores the sharp, scale-invariant reasoning absent from softmax, over NP-hard/complete tasks.

Lean Method Zoo – Paradigms to Scale Formal Proving

Fabien Glöckle

I presented different methods to use language models with interactive theorem proving such as Lean. These methods include fine-level tree search, and reinforcement learning extensions: MCTS, whole-proof generation and its iterative variants (multi-turn debugging), natural language conditional and hierarchical decomposition (Lean-conjecturing) variants.

For each of the methods, we make an attempt at analyzing its scaling behavior, because the ultimate question is: which paradigm shall we use to scale large-scale reinforcement learning for Lean ?

Why Autoformalization Will Be Both Feasible and Necessary?

Christian Szegedy

Here we gave a motivation and overview of autoformalization as a promising path towards making a superhuman AI mathematician. My presentation explained how autoformalization will give rise to necessary training data for verifiable mathematical reasoning, and also demonstrate the feasibility of autoformalization by the possibility of autoformalization by example of the Gauss system that was successfully employed to formalize the classical prime number theorem by Hadamard and de la Vallée Poussin.

Reporter: Antoine Peyronnet

Participants

Prof. Dr. Jeremy Avigad
Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA 15213-3890
UNITED STATES

Prof. Dr. Matthew R. Ballard
Department of Mathematics
University of South Carolina
1523 Greene St.
Columbia, SC 29208
UNITED STATES

Dr. Barinder Banwait
Department of Mathematics and Statistics
Boston University
One Oxford Street
Boston MA 02138-2901
UNITED STATES

Prof. Dr. Gergely Bérczi
Ny Munkegade 118
Matematisk Institut
Aarhus Universitet
8000 Aarhus C
DENMARK

Prof. Dr. François Charton
FAIR
Meta, Paris
6 rue menars
75002 Paris
FRANCE

Dr. Alexander Chervov
Institut Curie
20 rue d'Ulm
57248 Paris cedex 05
FRANCE

Dr. Edgar Costa
Department of Mathematics
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139-4307
UNITED STATES

Dr. María Inés de Frutos-Fernández
Mathematisches Institut
Universität Bonn
Endenicher Allee 62
53115 Bonn 53115
GERMANY

Dr. Jan de Gier
School of Mathematics and Statistics
The University of Melbourne
Parkville VIC 3010
AUSTRALIA

Prof. Dr. Michael R. Douglas
CMSA
Harvard University
20 Garden St
Cambridge MA 02138-2901
UNITED STATES

Prof. Dr. Romuald Elie
8 rue de Londres,
Google Deepmind
Tassigny
75009 Paris Cedex 16
FRANCE

Fabian Glöckle
ENPC-CERMICS
Champs-sur-Marne
6 av. Blaise Pascal
77455 Marne-la-Vallée
FRANCE

Javier Gomez-Serrano
Department of Mathematics
Brown University
Box 1917
Providence, RI 02912
UNITED STATES

Prof. Dr. William Timothy Gowers
Department of Pure Mathematics
and Mathematical Statistics
University of Cambridge
Wilberforce Road
Cambridge CB3 0WB
UNITED KINGDOM

Alex Gu
MIT CSAIL
400 East Remington Drive
P.O. Box C307
Sunnyvale CA 94087
UNITED STATES

Baran Hashemi
ORIGINS Cluster
TU München
Boltzmannstr. 2
85747 Garching bei München
GERMANY

Prof. Dr. Amaury Hayat
CERMICS – Ecole des Ponts
Institut Polytechnique de Paris
6-8 avenue Blaise Pascal
77420 Champs-sur-Marne Cedex 05
FRANCE

Carina Letong Hong
Axiom Math
Stanford, CA 94305-2125
UNITED STATES

Dr. Albert Jiang
Mistral AI
8 Cross Street
Cambridge CB1 2HW
UNITED KINGDOM

Prof. Dr. Julia Kempe
Courant Institute of
Mathematical Sciences
New York University
251, Mercer Street
New York, NY 10012-1110
UNITED STATES

Dr. Alexandre Krajenbrink
Quantinuum
Carlisle Pl
SW1P 1BX London
UNITED KINGDOM

Prof. Dr. Kyu-Hwan Lee
Department of Mathematics
University of Connecticut
Storrs CT 06269-1009
UNITED STATES

Prof. Dr. Heather Macbeth
Department of MathematicsImperial
College London
180 Queen's Gate, Huxley Bldg.
London SW7 2BZ
UNITED KINGDOM

Jonas Nehme
Mathematisches Institut
Universität Bonn
Endenicher Allee 60
53115 Bonn
GERMANY

Prof. Dr. Thomas Oliver
University of Westminster
New Cavendish Street
London W1W 6UW
UNITED KINGDOM

Dr. Antoine Peyronnet
Laboratoire Cermics
Cité Descartes
8 avenue Blaise Pascal 6
77420 Champs-sur-Marne
FRANCE

Maria Prat Colomer
Division of Applied Mathematics
Brown University
Box F
Providence, RI 02912
UNITED STATES

Prof. Dr. Catharina Stroppel
Mathematisches Institut
Universität Bonn
Endenicher Allee 60
53115 Bonn 53115
GERMANY

Andrew Sutherland
Department of Mathematics
Massachusetts Institute of Technology
77 Massachusetts Avenue
Cambridge, MA 02139-4307
UNITED STATES

Dr. Christian Szegedy
1450 Page Mill Road
Palo Alto CA 94304
UNITED STATES

Prof. Dr. Floris van Doorn
Mathematisches Institut
Universität Bonn
Endenicher Allee 60
53115 Bonn 53115
GERMANY

Dr. Adam Zsolt Wagner
Google DeepMind
6 Pancras Square
London N1C 4AG
UNITED KINGDOM

Prof. Dr. Melanie Weber
Harvard University
Massachusetts Hall
Cambridge MA 02138
UNITED STATES

Prof. Dr. Anna Katharina Wienhard
Max-Planck-Institut für Mathematik
in den Naturwissenschaften
Inselstr. 22 - 26
04103 Leipzig
GERMANY

Prof. Dr. Geordie Williamson
Sydney Mathematical Research Institute
School of Mathematics and Statistics
Faculty of Science
Quadrangle A14
The University of Sydney
Sydney NSW 2006
AUSTRALIA

Prof. Dr. Melanie Matchett Wood
Department of Mathematics
Harvard University
Cambridge, MA 02138
UNITED STATES