VerifAI-2: The Second Workshop on AI Verification in the Wild

1 Workshop Summary

This workshop series explores the intersection of scale-driven generative artificial intelligence (AI) and the correctness-focused principles of verification. In its first rendition at ICLR 2025, it focused in particular on how generative AI could address the scaling challenges faced by formal analysis tools such as theorem provers, satisfiability solvers, and execution monitoring. The special theme of VerifAI@ICLR'25 was thus Large Language Models (LLMs) for Code Generation, an undeniably active area of research across both industry and academia which has benefited greatly from (and improved) formal analysis tools such as static analyzers. Now, in light of the recent emphasis on large-scale post-training through reinforcement learning (RL), we are excited to continue uniting the interests of industry and academia with a new special theme: Building verifiable tasks and environments for RL. Potential topics thus include, but are not limited to, the following:

- **Generative AI for formal methods:** Formal methods offer strong guarantees of desired or undesired properties, but they can be challenging to implement. When faced with non-halting proofs or extensive search spaces, machine learning approaches can help guide those search processes effectively [7, 17] or even write the theorems themselves [3, 16]. How can we further integrate AI to enhance verification practices? How can we ensure that AI-generated test conditions [14, 1] align with actual desired properties?

- **Formal methods for generative AI:** Generative AI can benefit from formal methods that provide assurances that build trust. For example, satisfiability solvers can be used as a bottleneck in reasoning domains [9, 11], code generated by the model can be annotated with specifications for program analysis tools to ensure its correctness [10, 18], and even simple symbolic methods such as automata simulators can steer AI generations towards more logically consistent behavior [21]. How else can we integrate formal methods into generative AI development and usage?

- **AI as verifiers:** Hard guarantees can be notoriously rigid and/or difficult to achieve. In these cases, probabilistic methods are appealing alternatives to provide "soft assurances" [22]. Language agents, for example, are increasingly used to provide verbal feedback [13] to other language models. How can we develop more robust and trustworthy verifiers from probabilistic methods? In what settings is it appropriate to make verification more flexible using probabilistic methods?

- **Datasets and benchmarks:** Advancing research at the intersection of generative AI and verifiable environments requires robust datasets and benchmarks. While domains such as code generation have well established single-turn benchmarks [2, 4, 6], how to best evaluate models in multi-turn environments remains an open problem. We welcome papers that present new datasets and benchmarks in reasoning, theorem proving, multi-turn code generation, and related areas. How can we design benchmarks that accurately reflect the challenges in combining probabilistic models with formal (or informal) verification?

- **Special Theme: Verifiable tasks and environments for RL.** Reinforcement Learning with Verifiable Rewards (RLVR) has recently become a very important part of large-scale post-training [5, 8], yet its scope has mostly been limited to easily verifiable settings such as generating isolated code fragments [15]. This year, our special theme invites researchers to explore how to extend the success of RLVR to other domains by constructing new tasks and environments for RL that put verification at their center. Specific topics of interest may include: developing more scalable and efficient programmatic verifiers [19], improving the reward density of verifiers in multi-turn interactions [20, 12], and analyzing how RLVR affects the broad reasoning capabilities of general-purpose foundation models.

We welcome novel methodologies, analytic contributions, works in progress, negative results, and review and positional papers that will foster discussion. We will also have a track for tiny or short papers.

2 Tentative schedule

The program consists of a series of six invited talks (25 minutes + 5 minutes QA), three highlighted talks (10 minutes) selected from submitted papers, and two poster sessions (1 hour each). New for this year, we will also feature a 1-hour mentoring session, in which senior researchers will be paired with those new to the field to offer advice and insights related to the topic. This will take the place of last year's panel discussion; the new, decentralized format will allow for more people to be involved, thus facilitating greater knowledge transfer. A tentative schedule is as follows:

Time	Event	Time	Event
8:55 - 9:00	Opening Remarks	13:00 - 13:30	Invited Talk 4
9:00 - 9:30	Invited Talk 1	1:30 - 2:00	Invited Talk 5
9:30 - 10:00	Invited Talk 2	2:00 - 2:30	Invited Talk 6
10:00 - 10:30	Invited Talk 3	2:30 - 3:00	Highlight Talks
10:30 - 11:00	Coffee Break	3:00 - 4:00	Poster Session 2
11:00 - 12:00	Poster Session 1	4:00 - 5:00	Mentoring Session
12:00 - 13:00	Lunch Break	5:00 - 5:05	Concluding Remarks

3 Invited speakers and panelists

Işıl Dillig, *Professor, University of Texas, Austin*, Website: **(confirmed)**: Işıl Dillig is a professor at the Computer Science Department of the University of Texas at Austin, where she leads the UToPiA research group. Her primary research area is Programming Languages (PL), with a current emphasis on program synthesis and verification. Her research has been recognized with several best or distinguished paper awards, such as at POPL'22, CHI'21, OOPSLA'20, PLDI'19, PLDI'18, OOPSLA'17, and ETAPS'17. She was also selected as a Sloan Fellow in 2015, won an NSF CAREER award in 2015, and was recently awarded the 2025 ACM SIGPLAN Robin Milner Award. Işıl obtained all of her degrees (BS, MS, and PhD) in Computer Science from Stanford University.

Gabriel Poesia, *Incoming Assistant Professor, University of Michigan*, Website: **(confirmed)**: Gabriel Poesia is an incoming Assistant Professor at the University of Michigan, a current Research Fellow at Harvard University's Kempner Institute, and a recent graduate of Stanford University. He works on building self-improving AI systems that are capable of formal reasoning and open-ended discovery. Gabriel's approach is inherently interdiscplinary, integrating ideas from type theory, reinforcement learning, language models, program induction, and the whole toolbox from game-playing AI. Outside of research, Gabriel is known for his service to the competitive programming community in his native Brazil and Latin America more broadly; he was an ACM-ICPC world finalist in 2015, has authored several problems for the ACM-ICPC Latin American regional competitions, and has coached several teams and training camps throughout the region.

Emily McMilin, *Research Scientist, Meta*, Website: **(in contact)**: Emily McMilin is currently a research scientist at Meta working on Language Modeling, Causal Inference, and Applied Reinforcement learning. Her recent work in world models for coding at FAIR builds upon a robust body of prior work in causal inference and biases, many contributions of which came as a single-author independent researcher.

Federico Mora Rocha, *Applied Scientist, Amazon Web Services; Incoming Assistant Professor, University of Waterloo*, Website: **(confirmed)**: Federico Mora Rocha is an Applied Scientist in the Automated Reasoning Group at Amazon Web Services, an incoming Assistant Professor at the University of Waterloo and a Faculty Affiliate at the Vector Institute. His research is centerd around automated reasoning, programming languages and their interactions with neurosymbolic systems. Federico obtained his PhD from the University of California, Berkeley in 2025; in addition to being awarded a Qualcomm Innovation Fellowship (2021) and winning the QF_Datatypes division of SMT-COMP (2024), Federico has been honored with several awards for his dedication to mentoring peers and undergraduate students at UC Berkeley.

Elizabeth Polgreen, *Assistant Professor, University of Edinburgh*, Website: **(in contact)**: Elizabeth Polgreen is an assistant professor at the University of Edinburgh. She is interested in formal program synthesis techniques and the use of synthesis to increase the scalability of verification. She holds a research fellowship from the Royal Academy of Engineering. Her work in lifting C code to a tensor DSL has received the Best Paper Award at GPCE.

Naman Jain, *PhD Student, UC Berkeley; Researcher, Cursor AI*, Website: **(confirmed)**: Naman Jain is a PhD student at UC Berkeley and researcher at Cursor AI. His research, focused on evaluation of and reinforcement learning environments for LLM coding agents, includes prominent benchmarks such as LiveCodeBench and an open-sourced framework that turns any GitHub repository into test environments for coding agents. He is now at Cursor putting his expertise into commercial-grade code development environments.

4 Organizers and biographies

The organizing committee of VerifAI-2 remains largely unchanged from that of VerifAI at ICLR 2025, with five out of six members of this year's team (Celine, Ameesh, Theo, Sean, and Tao) having also organized the previous instantiation of the workshop. The team is rounded out by Armando Solar-Lezama, whose prominence in the programming languages community will help align the workshop with the goals and interests of a broader audience.

Celine Lee
PhD Student, Cornell
cl923@cornell.edu
Website, Google Scholar
Celine Lee is a PhD candidate at Cornell University, advised by Sasha Rush. Her research focuses on studying the capabilities of LLMs for low-level code and symbolic reasoning. Her work has been presented at top-tier conferences such as ICLR, ACL, EMNLP, and KDD. She has interned as a researcher at Google Deepmind, Intel Labs and IBM Research, and holds several patents in AI-supported program synthesis and code management. She has taught as an adjunct instructor at MBZUAI, and is recognized as a 2025 MLSys Rising Star.

Ameesh Shah
PhD Student, UC Berkeley
ameesh@berkeley.edu
Website, Google Scholar
Ameesh Shah is a Ph.D. candidate in Computer Science at UC Berkeley, advised by Sanjit Seshia. His research focuses broadly at the intersection of machine learning and formal methods, with aims of building assurances for learning-enabled systems through formal logic and verification. His work, which spans across program synthesis, reinforcement learning, and robotics, has appeared at top venues in ML and formal methods including NeurIPS, ICML, ICLR, AAMAS, and FMCAD. He is the recipient of the NDSEG Fellowship and has interned with groups at Microsoft Research and Toyota Research Institute concerning safe and trustworthy ML.

Theo X. Olausson
PhD Student, Massachusetts Institute of Technology
theoxo@mit.edu
Website, Google Scholar
Theo X. Olausson is a PhD candidate at the Massachusetts Institute of Technology, where he is advised by Armando Solar-Lezama. His research focuses on improving the reliability and scalability of LLMs and other foundation models, often by integrating symbolic tools such as interpreters or SAT solvers into the loop. Theo's work has appeared at conferences such as ICLR, ACL, POPL, and EMNLP, and was recognized with an Outstanding Paper award at EMNLP'23 as well as a Presidential Fellowship from MIT. He has interned with the Deep Learning group at Microsoft Research, the Simons Foundation's Center for Computational Mathematics, and Apple Machine Learning Research.

Sean Welleck
Assistant Professor, CMU
wellecks@cmu.edu
Website, Google Scholar

Sean Welleck is an Assistant Professor at Carnegie Mellon University, where he leads the Machine Learning, Language, and Logic (L3) Lab. His areas of focus include generative models, algorithms for large language models, and AI for code and mathematics. Sean received a Ph.D. from New York University. He was a postdoctoral scholar at the University of Washington and AI2. In addition to seving on organizing committee of VerifAI at ICLR 2025, Sean has co-organized several workshops on AI + Math (Math AI for Education, NeurIPS 2021; Math AI, Neurips 2022; Math AI, Neurips 2023; AI for Math, ICML 2024).

Armando Solar-Lezama
Distinguished Professor of Computing, Massachusetts Institute of Technology
asolar@csail.mit.edu
Website, Google Scholar

Armando Solar-Lezama is the Distinguished Professor of Computing at MIT's Schwarzman College of Computing. He is also Associate Director and COO of MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL). The focus of Armando's research is program synthesis, an exciting area that lies at the intersection of Programming Systems and Artificial Intelligence. Over the past five years, he and his students have also been working on developing neurosymbolic programming, a new class of learning techniques that incorporate some of the benefits of traditional programming languages, such as modularity, compositionality, and predictability. Armando obtained his BS degrees in Computer Science and Mathematics from Texas A&M University before moving on to the University of California, Berkeley for his PhD. He has been a member of the faculty at MIT since 2008.

Tao Yu
Assistant Professor, The University of Hong Kong
tao.yu.nlp@gmail.com
Website, Google Scholar

Tao Yu is an Assistant Professor at The University of Hong Kong. He completed his Ph.D. in Computer Science from Yale University. His research aims to build language model agents that transform ("grounding") language instructions into code or actions executable in real-world environments, including databases, web applications, and the physical world. Tao has received the Google and Amazon faculty research awards (Google Research Scholar Award 2023, Amazon Research Award 2022). Tao has served on the organizing committee of ACL 2023, SUKI: Structured and Unstructured Knowledge Integration Workshop@NAACL 2022, and IntEx-SemPar: Interactive and Executable Semantic Parsing Workshop@EMNLP 2020.

5 Marketing and anticipated audience size

We will create and maintain a workshop website advertising the topic, call for papers, speaker list, and tentative schedule; this will be similar in design and scope to that of VerifAI at ICLR 2025. We will promote the workshop website and call-for-papers through social media platforms, email lists, and research communities to reach researchers from various disciplinary and biographic backgrounds.

We aim for our workshop to attract researchers from the machine learning and programming languages community alike. The number of research papers appearing at ICLR that pertain to code generation and code reasoning continues to grow, and with the renewed focus of RLVR we expect to attract an ever wider audience than in 2025. Based on these factors and the attendance levels of this year's edition, we anticipate that ~ 200 attendees will participate in this workshop.

6 Diversity commitment

Diversity of organizers and speakers. Despite its small size, our organizing committee spans multiple universities (Cornell, Berkeley, MIT, HKU, and CMU) and levels of seniority (PhD students to distinguished professors). Furthermore, the team members cover a diverse expertise in machine learning, language modeling, code generation, reasoning, executable language grounding, formal methods, program synthesis, neurosymbolic AI, code verification, theorem proving, and evaluation.

We have also invited six distinguished speakers, which have ben selected to represent a wide range of perspectives on the workshop topic. Specifically, we have taken special care to select speakers that span the full width of the formal verification to mainstream AI spectrum, as well as representing a mix between academic and industry labs.

Between the organizing committee and speakers, four out of twelve members of the workshop team are female researchers. The home institutions of our team cover North America, parts of Europe and parts of Asia. In addition, with ICLR 2026's location in mind, we are especially happy to feature several Latinx members in the team this year to promote exchange between local and remote research communities.

Diversity of participants. We plan to ensure the presence of all related communities by broadcasting our workshop to faculty, researchers, and students working in these areas. When calling for participation, we will refer to the BIG Directory[1] to encourage submissions/participation from the underrepresented groups. We will use part of any external funding we secure to support such participants.

7 Access

The workshop will be primarily held in person, but to accommodate those who cannot physically attend, we will maintain and share a virtual livestream through which remote attendees can

[1] http://www.winlp.org/big-directory/

participate. At all times, an organizer will monitor the stream for questions directed to the the organizers and/or speakers. All accepted papers, talks, and discussions will also be posted to our aforementioned website for perusal during and after the workshop.

8 Previous related workshops

This Second Workshop on AI Verification in the Wild builds on the success of the first rendition, VerifAI at ICLR 2025. In addition, there are several other long-running running workshop series that discuss related topics, albeit with slightly different goals:

- Deep Learning for Code (DL4C)
- LLMs for Code (LLM4Code)
- Mathematical Reasoning and AI (MATH-AI)
- ML for Systems
- Workshop on Formal Verification and Machine Learning (WFVML)
- Symposium on AI Verification (SAIV)

These workshops have fostered conversation about deep learning methods for code, computer systems, software engineering (e.g. DL4C, LLM4Code, ML for Systems) and mathematic reasoning (e.g. MATH-AI) but without a particular focus on verification or the interaction between agentic AI and verifiable environments. Conversely, other workshops have focused on formal verification for and with a broad range of AI tools (e.g. WFVML, SAIV), but do not highlight opportunities newly available in the context of contemporary advances such as LLMs and RLVR. With VerifAI-2: The Second Workshop on AI Verification in the Wild, we are excited to continue building on the momentum initiated by VerifAI at ICLR 2025, while responding to the challenges that have become most relevant to the community in the year since by designating RLVR this edition's special theme.

References

[1] N. Alshahwan, J. Chheda, A. Finogenova, B. Gokkaya, M. Harman, I. Harper, A. Marginean, S. Sengupta, and E. Wang. Automated unit test improvement using large language models at meta. In *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, FSE 2024, page 185–196, New York, NY, USA, 2024. Association for Computing Machinery.

[2] J. Austin, A. Odena, M. Nye, M. Bosma, H. Michalewski, D. Dohan, E. Jiang, C. Cai, M. Terry, Q. Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[3] Z. Azerbayev, H. Schoelkopf, K. Paster, M. D. Santos, S. M. McAleer, A. Q. Jiang, J. Deng, S. Biderman, and S. Welleck. Llemma: An open language model for mathematics. In *The Twelfth International Conference on Learning Representations*, 2024.

[4] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating large language models trained on code. 2021.

[5] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[6] N. Jain, K. Han, A. Gu, W.-D. Li, F. Yan, T. Zhang, S. Wang, A. Solar-Lezama, K. Sen, and I. Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint*, 2024.

[7] A. Q. Jiang, S. Welleck, J. P. Zhou, T. Lacroix, J. Liu, W. Li, M. Jamnik, G. Lample, and Y. Wu. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. In *The Eleventh International Conference on Learning Representations*, 2023.

[8] N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, S. Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[9] T. X. Olausson*, A. Gu*, B. Lipkin*, C. E. Zhang*, A. Solar-Lezama, J. B. Tenenbaum, and R. P. Levy. Linc: A neuro-symbolic approach for logical reasoning by combining language models with first-order logic provers. 2023.

[10] K. Pei, D. Bieber, K. Shi, C. Sutton, and P. Yin. Can large language models reason about program invariants? In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[11] G. Poesia, K. Gandhi, E. Zelikman, and N. D. Goodman. Certified reasoning with language models. 2023.

[12] R. Shao, S. S. Li, R. Xin, S. Geng, Y. Wang, S. Oh, S. S. Du, N. Lambert, S. Min, R. Krishna, et al. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

[13] N. Shinn, F. Cassano, A. Gopinath, K. R. Narasimhan, and S. Yao. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[14] M. L. Siddiq, J. C. Da Silva Santos, R. H. Tanvir, N. Ulfat, F. Al Rifat, and V. Carvalho Lopes. Using large language models to generate junit tests: An empirical study. In *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, EASE '24, page 313–322, New York, NY, USA, 2024. Association for Computing Machinery.

[15] Y. Wang, Q. Yang, Z. Zeng, L. Ren, L. Liu, B. Peng, H. Cheng, X. He, K. Wang, J. Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025.

[16] S. Welleck, J. Liu, X. Lu, H. Hajishirzi, and Y. Choi. Naturalprover: Grounded mathematical proof generation with language models. In A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[17] S. Welleck and R. Saha. Llmstep: Llm proofstep suggestions in lean. *arXiv preprint arXiv:2310.18457*, 2023.

[18] C. Yang, X. Li, M. R. H. Misu, J. Yao, W. Cui, Y. Gong, C. Hawblitzel, S. Lahiri, J. R. Lorch, S. Lu, F. Yang, Z. Zhou, and S. Lu. Autoverus: Automated proof generation for rust code, 2024.

[19] T. Yu, B. Ji, S. Wang, S. Yao, Z. Wang, G. Cui, L. Yuan, N. Ding, Y. Yao, Z. Liu, et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv preprint arXiv:2506.18254*, 2025.

[20] S. Zeng, Q. Wei, W. Brown, O. Frunza, Y. Nevmyvaka, and M. Hong. Reinforcing multi-turn reasoning in llm agents via turn-level credit assignment. *arXiv preprint arXiv:2505.11821*, 2025.

[21] H. Zhang, P.-N. Kung, M. Yoshida, G. V. den Broeck, and N. Peng. Adaptable logical control for large language models, 2024.

[22] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc., 2023.