Advances in Programming Languages and Neurosymbolism (AIPLANS II)

Jialu Bao[3], **Breandan Considine**[1,11], **Maddy Bowers**[2],
Younesse Kaddar[5], **Justine Gehring**[9,12], **Shawn Tan**[4,11]
PC: Swarat Chaudhuri[10], David Chiang[6], Parisa Kordjamshidi[8],
Nikolay Malkin[9,11], Xujie Si[7,13], Sam Staton[5]
McGill[1], MIT[2], Cornell[3], Université de Montréal[4], Oxford[5], Notre Dame[6],
University of Toronto[7], Michigan State University[8], University of Edinburgh[9],
Unviersity of Texas at Austin[10], Mila[11], Moderne[12], Vector[13]

Tagline: AIPLANS: Fusing machine learning with programming language theory to create neurosymbolic programming machines! `https://aiplans.github.io`

Since the first AIPLANS workshop at NeurIPS 2021, rapid progress has been made in uniting two divergent branches of computer science: machine learning (ML) and programming languages (PL). What was once a fringe research area has begun to catalyze a broader movement in neurosymbolic methods, with summer schools [38, 11], workshops [45, 3, 37, 32] and tutorials [39, 7, 43] dedicated to this field of research in recent years. From the connectionists descended the present generation of large language models, inspired by biological systems and grounded in probabilistic modeling and optimization; from the symbolists came programming languages, compilers, and automated theorem provers, powering the digital revolution which made the endeavor of AI possible.

AIPLANS seeks to harness the emergent capabilities of foundation models along with the explainability and rigor of symbolic methods, taking inspiration from experimentalists, language designers, programming enthusiasts, and formalists alike. How do we build systems with the acumen of large language models while retaining the safety and trustworthiness of programming languages? Can we make them flexible enough to behave naturally while compositional enough to be understood? And how do we guarantee these systems plan and act safely [23] amidst humans? At AIPLANS, we aim to explore these questions with an eye towards *program synthesis* – the generation of interpretable [15], mechanically verifiable programs. In a series of invited talks, contributed papers, and panel discussions, we plan to cover some of the following areas:

- Probabilistic model checking [27]
- Language induction
- Proof search and program synthesis
- Declarative programming [26]
- Constraint satisfaction
- Unification, completion and resolution
- Analytic combinatorics [17]
- Computational group theory
- Vector symbolic architectures [22]
- Automata and formal grammars
- Formal aspects of language models [12]
- Constrained sampling from LLMs
- Randomized complexity
- Algorithmic information theory [6]
- Circuit lower bounds [34]
- Neural abstract interpretation [36]
- Proof assistance and automation
- Type theory and semantics

The goal of AIPLANS is to provide a forum for researchers working in programming languages, machine learning, and automated reasoning to share novel insights on generating discrete combinatorial structures like proofs and programs, as well as characterizing the logical expressivity of existing machine learning models. By specifically targeting program synthesis, we hope to bring a more formal approach to machine learning and foster increased collaboration between the PL and ML communities. We believe that NeurIPS is a uniquely well-suited venue to hold our workshop.

38th Conference on Neural Information Processing Systems (NeurIPS 2024), Vancouver, Canada.

While other prior workshops have focused on generating source code [42, 14], AIPLANS is more devoted to applying the theory of formal languages to understand machine learning systems, as well as synthesizing interpretable and verifiable programs. Rather than treating programs as ordinary strings, we can leverage a rich body of PL literature to test their soundness and prove behavioral properties. In parallel, an emerging body of neurosymbolic literature is starting to explore the computational expressivity of LLMs [44], languages for tractable [10] and intractable [24] inference, the learnability of formal languages [29], and evaluating the systematic generalization of language models on tasks of increasing complexity such as program [20, 18], logic [50], and automata [31, 41] simulation.

Related Literature

Our workshop is particularly interested in exploring the intersection of the following research areas.

Probabilistic programming

Probabilistic programming languages (PPLs) are domain-specific languages (DSLs) for expressing probabilistic models and inference algorithms, performing stochastic computation and reasoning about uncertainty [5, 4, 13, 19, 48]. They typically implement common operations such as generating pseudorandom numbers, estimating probability density functions, and manipulating them by conditioning, sampling, marginalizing [30] and calculating expectations and higher moments [35]. More sophisticated PPLs [9] enable users to prove higher-order properties of probabilistic programs [25, 51] such as rates of convergence, concentration bounds [33] and amortized analysis of algorithms [2].

Program synthesis

Program synthesis is broadly interested in generating programs that satisfy a specification, whether from a natural language description [1], input-output examples [21], or logical formulae. This area of research is closely related to language induction and shares close ties with formal methods, probabilistic programming, and symbolic regression [47]. The goal of program synthesis is to automate the process of programming and to generate programs that are correct by construction.

Formal language theory

The theory of computation offers a foundation for programming language and machine learning research. Formal language theory explores what is and is not possible, how hard it will be, and if one poses the right questions, a constructive mechanism for how to realize it in silico. The tools it provides rigorously characterize the properties of languages and computing machinery, such as their expressivity [8], learnability [29], and automatizability. This is a promising area of research for understanding the limits of what can be learned and computed and is a cornerstone of AIPLANS.

Neurosymbolism

There is a virtuous cycle between learning and reasoning that is neglected in purely data-driven learning. Neurosymbolism seeks to distill knowledge from the world and enrich the learning process by incorporating symbolic knowledge from prior experience. This can take the form of programmatic or inductive biases that a domain expert supplies [16], or the extraction of latent knowledge from trained neural networks [49] into symbolic form, to admit e.g., interpretation or tractable verification. Another large body of neurosymbolic research studies symbol grounding [46], as well as language games, pragmatic communication [40], and abductive reasoning.

Speakers

AIPLANS has invited each of the following speakers and received written confirmation from those marked ✓ they will attend in person, contingent on workshop acceptance:

✓ Parisa Kordjamshidi ? Ekaterina Komendantskaya ? Conal Elliott ? Will Merrill

Proposed Workshop Logistics

AIPLANS will be a one-day in-person workshop with live talks and panels. Talks will be hosted in English, following the standard format of oral presentations and panel discussions, to be concluded with a poster session. Proceedings will be non-archival. Outside of standard videoconferencing and SlidesLive assistance, no other technical requirements are necessary. We anticipate receiving roughly 200 participants, including speakers and workshop submitters, based on our experience running AIPLANS at NeurIPS 2021 and attendance at similarly-themed workshops in prior years.

The workshop itself will run for approximately eight hours, featuring four 45-minute contributed talks and up to six 20-minute contributed talks, with a strong preference for in-person attendance. AIPLANS will solicit four-page paper submissions in a CFP to be circulated pending workshop acceptance. To encourage submissions from the broader ML/PL community, accepted authors will be given an opportunity to showcase their work in a poster session, and exceptional contributions may be selected for a short spotlight talk. We expect to receive 30-40 submissions, and pledge that each paper will receive at least two fair and independent reviews. To minimize potential conflicts of interest, AIPLANS will manage submissions via OpenReview.

This year, AIPLANS is piloting a new format for contributed work. In keeping with Leslie Lamport's advice [28], submissions will be asked to (1) explicitly state the problem they are trying to solve, then (2) describe how they recognize success. This can take the form of a proof or empirical evidence.

Those who traditionally publish in venues such as SIGPLAN, SIGSOFT and other ACM venues are encouraged to submit work they consider relevant to a machine learning audience, provided that effort is taken to ensure its accessibility. Special consideration will be given to pedagogical submissions of outstanding clarity. Further information, including evaluation criteria, examples of relevant literature, deadlines and workshop logistics will be provided for reference.

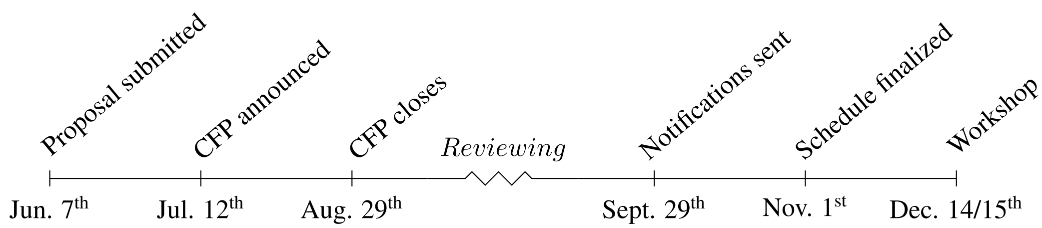
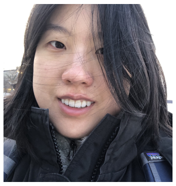


Figure 1: A tentative timeline for our proposed workshop at NeurIPS 2024.

If accepted, AIPLANS will announce its CFP and pursue contributions from the broader ML/PL community shortly thereafter. Six weeks later, the CFP will close on Aug. 29th. This deadline may be extended to no later than Sept. 5th, depending on the volume of submissions received, leaving sufficient time for referees and program chairs to give feedback. Authors will be notified of acceptance no later than Sept. 29th. We intend to finalize the schedule and coordinate the presentation logistics between Nov. 1st and Dec. 14th. Those who wish to prerecord their talk or present remotely will be permitted to do so under extenuating circumstances, up to a limit of one hour as per conference guidelines. The final workshop will consist primarily of live talks with Q&A, followed by a moderated panel, and poster session. Further details about schedule and logistics will be made available, pending acceptance at: `https://aiplans.github.io`.

AIPLANS is an equal-opportunity workshop that celebrates cultural, linguistic, ethnic and intellectual diversity in all forms. Not only are we committed to nondiscrimination on the basis of, e.g., race, creed, age, gender, orientation, physical or mental handicap, but also aim to encourage individuals from other disadvantaged and underrepresented socioeconomic backgrounds to participate. Should our workshop be accepted, scholarships covering the cost of registration will be extended for those who wish to attend but would otherwise be unable to do so due to financial hardship. If needed, AIPLANS may pursue industry sponsorship for this initiative to enable a wider audience to attend. Further details about registration and funding availability will be provided in a timely manner.

Confirmed Workshop Organizers



Jialu Bao is a Ph.D. student at Cornell advised by Prof. Justin Hsu, who is working on the verification of randomized algorithms. Before moving to Cornell with her advisor, she spent two years at University of Wisconsin – Madison as a Ph.D. student, and prior to that, did her undergrad at Cornell majoring in Mathematics and Computer Science.

`https://baojia.lu` @howowhy

Breandan Considine is a Ph.D. student at McGill University co-supervised by Jin Guo and Xujie Si. His research studies how to reason about the behavior of real-world programs and build more intelligent programming tools for developers. Previously, he organized the first AIPLANS workshop at NeurIPS and co-organized the ICLR workshop, Rethinking ML Papers.

`https://breandan.github.io` @breandan

Maddy Bowers is a Ph.D. student at MIT, co-advised by Armando Solar-Lezama in EECS and Josh Tenenbaum in BCS. Her research combines methods from programming languages (PL) research with machine learning to tackle problems in artificial intelligence.

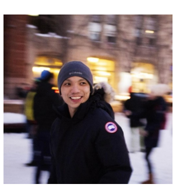
`https://mlb2251.github.io/` @mattlbowers

Younesse Kaddar is a Ph.D. student in theoretical computer science at the University of Oxford, working on programming language semantics, Bayesian probabilistic programming, and category theory. Previously, he was a visiting researcher at Mila/IQIA working with Yoshua Bengio.

`https://younesse.net/` @you_kad

Justine Gehring is a research engineer at Moderne working in the field of Machine Learning (ML) for code and Graph Neural Networks (GNNs). Her focus lies in generating code under challenging circumstances, specifically in scenarios such as sparse data where library-specific code is required, as well as managing a substantial amount of code at a time. She completed her Master's Degree in Computer Science at Mila & McGill.

`https://justine-gehring.github.io/` @GehringJustine



Shawn Tan is a Ph.D. candidate at Mila, Université de Montréal. He is interested in differentiable methods for structured prediction, specifically in the domain of natural language. He co-authored the Ordered Neurons paper which won best paper at ICLR 2019.

`http://blog.wtf.sg` @tanshawn

Confirmed Program Committee

Swarat Chaudhuri is a Professor of Computer Science and leads the Trishul lab at UT Austin. His research lies at the interface of programming languages, formal methods, and machine learning. His aim as a researcher is to develop a new class of intelligent systems that are reliable, transparent, and secure by construction and can solve reasoning-intensive tasks beyond the scope of contemporary AI.

`https://www.cs.utexas.edu/~swarat/` @swarat

David Chiang is an associate Professor at the University of Notre Dame. His research is in natural language processing, machine learning, and programming languages, particularly in applications of formal language theory to these areas.

`https://www3.nd.edu/~dchiang/` @davidweichiang

Parisa Kordjamshidi is an assistant professor at Michigan State University and director of the Heterogeneous Learning and Reasoning (HLR) lab. Her main research interests are artificial intelligence, machine learning, natural language processing, and declarative learning based programming (DeLBP). She works on the extraction of formal semantics and structured representations from natural language, with a specific focus on spatial semantics representation and structured output learning models.

`https://www.cse.msu.edu/~kordjams/` @Kordjamshidi

Nikolay Malkin is a Chancellor's Fellow at the University of Edinburgh working on algorithms for deep-learning-based reasoning and their applications, in particular induction of compositional structure in generative models, modeling of posteriors over high-dimensional explanatory variables, and uncertainty-aware explanations for observed data, with the goal of automating human-like symbolic, formal, and mathematical reasoning.

`https://malkin1729.github.io/`

Xujie Si is an Assistant Professor and Canada CIFAR AI Chair in the School of Computer Science at the University of Toronto and Vector Institute. He finished his Ph.D. in Computer and Information Science at the University of Pennsylvania in 2020, advised by Prof. Mayur Naik. Xujie received his M.S. in computer science from Vanderbilt University in 2014, before which he obtained his B.E. (with Honors) from Nankai Unversity in 2011.

`https://www.cs.mcgill.ca/~xsi` @XujieSi

Sam Staton is a Professor of Computer Science at the University of Oxford. His research focuses on programming language theory, logic, and category theory, with an emphasis on probabilistic programming languages. These languages specify and understand statistical models through programming. Sam's recent work includes developing the LazyPPL library, working on semantics with s-finite kernels and quasi-Borel spaces, and exploring exchangeability and program modules.

`https://www.cs.ox.ac.uk/people/sam.staton/`

References

[1] Sam Acquaviva, Yewen Pu, Marta Kryven, Theodoros Sechopoulos, Catherine Wong, Gabrielle Ecanow, Maxwell Nye, Michael Tessler, and Josh Tenenbaum. Communicating natural programs to humans and machines. *Advances in Neural Information Processing Systems*, 35:3731–3743, 2022.

[2] Kevin Batz, Benjamin Lucien Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Lena Verscht. A calculus for amortized expected runtimes. *Proceedings of the ACM on Programming Languages*, 7(POPL):1957–1986, 2023.

[3] Vaishak Belle, Michael Fisher, et al. Neuro-symbolic AI for agent and multi-agent systems workshop. 2023.

[4] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul A. Szerlip, Paul Horsfall, and Noah D. Goodman. Pyro: Deep universal probabilistic programming. *J. Mach. Learn. Res.*, 20:28:1–28:6, 2019.

[5] Bob Carpenter, Andrew Gelman, Matthew D. Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1):1–32, 2017.

[6] Gregory J Chaitin. Algorithmic information theory. *IBM journal of research and development*, 21(4):350–359, 1977.

[7] Swarat Chaudhuri, Atharva Sehgal, and Armando Solar-Lezama. Tutorial on neuro-symbolic programming. 2023.

[8] David Chiang, Peter Cholak, and Anand Pillay. Tighter bounds on the expressivity of transformer encoders. In *International Conference on Machine Learning*, pages 5544–5562. PMLR, 2023.

[9] David Chiang, Colin McDonald, and Chung-chieh Shan. Exact recursive probabilistic programming. *Proceedings of the ACM on Programming Languages*, 7(OOPSLA1):665–695, 2023.

[10] Y Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA*, page 6, 2020.

[11] Omar Costilla Reyes et al. Summer school on neurosymbolic programming. 2024.

[12] Ryan Cotterell, Anej Svete, Clara Meister, Tianyu Liu, and Li Du. Formal aspects of language modeling. *arXiv preprint arXiv:2311.04329*, 2023.

[13] Marco F. Cusumano-Towner, Feras A. Saad, Alexander K. Lew, and Vikash K. Mansinghka. Gen: a general-purpose probabilistic programming system with programmable inference. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation*, PLDI 2019, page 221–236, New York, NY, USA, 2019. Association for Computing Machinery.

[14] Prem Devanbu, Yiling Lou, Lin Tan, and Lingming Zhang. First international workshop on large language models for code. 2024.

[15] Kevin Ellis, Lionel Wong, Maxwell Nye, Mathias Sable-Meyer, Luc Cary, Lore Anaya Pozo, Luke Hewitt, Armando Solar-Lezama, and Joshua B Tenenbaum. Dreamcoder: growing generalizable, interpretable knowledge with wake–sleep bayesian program learning. *Philosophical Transactions of the Royal Society A*, 381(2251):20220050, 2023.

[16] Hossein Rajaby Faghihi, Quan Guo, Andrzej Uszok, Aliakbar Nafar, Elaheh Raisi, and Parisa Kordjamshidi. Domiknows: A library for integration of symbolic domain knowledge in deep learning. *arXiv preprint arXiv:2108.12370*, 2021.

[17] Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. Cambridge University Press, 2009.

[18] Dan Friedman, Alexander Wettig, and Danqi Chen. Learning transformer programs. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] Hong Ge, Kai Xu, and Zoubin Ghahramani. Turing: A language for flexible probabilistic inference. 84:1682–1690, 09–11 Apr 2018.

[20] Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *International Conference on Machine Learning*, pages 11398–11442. PMLR, 2023.

[21] Sumit Gulwani. Automating string processing in spreadsheets using input-output examples. *ACM Sigplan Notices*, 46(1):317–330, 2011.

[22] Michael Hersche, Mustafa Zeqiri, Luca Benini, Abu Sebastian, and Abbas Rahimi. A neuro-vector-symbolic architecture for solving raven's progressive matrices. *Nature Machine Intelligence*, 5(4):363–375, 2023.

[23] Alasdair Hill, Ekaterina Komendantskaya, Matthew L Daggitt, and Ronald PA Petrick. Actions you can handle: dependent types for ai plans. In *Proceedings of the 6th ACM SIGPLAN International Workshop on Type-Driven Development*, pages 1–13, 2021.

[24] Edward J Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing intractable inference in large language models. *arXiv preprint arXiv:2310.04363*, 2023.

[25] Lutz Klinkenberg, Christian Blumenthal, Mingshuai Chen, Darion Haase, and Joost-Pieter Katoen. Exact bayesian inference for loopy probabilistic programs using generating functions. *Proceedings of the ACM on Programming Languages*, 8(OOPSLA1):923–953, 2024.

[26] Parisa Kordjamshidi, Dan Roth, and Kristian Kersting. Declarative learning-based programming as an interface to ai systems. *Frontiers in artificial intelligence*, 5:755361, 2022.

[27] Marta Kwiatkowska, Gethin Norman, and David Parker. Probabilistic model checking: Advances and applications. *Formal System Verification: State-of the-Art and Future Trends*, pages 73–121, 2018.

[28] Leslie Lamport. State the problem before describing the solution. *ACM SIGSOFT Software Engineering Notes*, 3(1):26–26, 1978.

[29] Nur Lan, Emmanuel Chemla, and Roni Katzir. Bridging the empirical-theoretical gap in neural network formal language learning using minimum description length. *arXiv preprint arXiv:2402.10013*, 2024.

[30] Anji Liu, Kareem Ahmed, and Guy Van den Broeck. Scaling tractable probabilistic circuits: A systems perspective. *arXiv preprint arXiv:2406.00766*, 2024.

[31] Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.

[32] Filip Llievski, Jacopo de Berardinis, Jongmo Kim, and Nitisha Jain. First international workshop on generative neuro-symbolic AI. 2024.

[33] Elisabet Lobo-Vesga, Alejandro Russo, and Marco Gaboardi. A programming framework for differential privacy with accuracy concentration bounds. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 411–428. IEEE, 2020.

[34] William Merrill, Ashish Sabharwal, and Noah A Smith. Saturated transformers are constant-depth threshold circuits. *Transactions of the Association for Computational Linguistics*, 10:843–856, 2022.

[35] Marcel Moosbrugger, Miroslav Stankovič, Ezio Bartocci, and Laura Kovács. This is the moment for probabilistic loops. *Proceedings of the ACM on Programming Languages*, 6(OOPSLA2):1497–1525, 2022.

[36] Mark Niklas Müller, Marc Fischer, Robin Staab, and Martin Vechev. Abstract interpretation of fixpoint iterators with applications to neural networks. *Proceedings of the ACM on Programming Languages*, 7(PLDI):786–810, 2023.

[37] Asim Munawar, Elham Barezi, et al. Neuro-symbolic learning and reasoning in the era of large language models workshop. 2024.

[38] Asim Munawar, Alexander Gray, et al. Neuro-symbolic AI summer school. 2023.

[39] Hamid Palangi, Antoine Bosselut, Pradeep Dasigi, et al. Neuro-symbolic methods for language and vision. 2022.

[40] Yewen Pu, Kevin Ellis, Marta Kryven, Josh Tenenbaum, and Armando Solar-Lezama. Program synthesis with pragmatic communication. *Advances in Neural Information Processing Systems*, 33:13249–13259, 2020.

[41] Michael Rizvi, Maude Lizaire, Clara Lacroce, and Guillaume Rabusseau. Simulating weighted automata over sequences and trees with transformers. *arXiv preprint arXiv:2403.09728*, 2024.

[42] Torsten Scholak, Gabriel Orlanski, Zijian Wang, Disha Shrivastava, and Devjeet Roy. Deep learning for code. 2023.

[43] Paulo Shakarian et al. Advances in neuro symbolic reasoning and learning. 2024.

[44] Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? a survey. *Transactions of the Association for Computational Linguistics*, 12:543–561, 2024.

[45] Thiviyan Thanapalasingam, Emile van Krieken, Halley Young, Disha Shrivastava, Jakub Tomczak, and Kevin Ellis. Neurosymbolic generative models workshop. 2023.

[46] Sever Topan, David Rolnick, and Xujie Si. Techniques for symbol grounding with SATNet. *Advances in Neural Information Processing Systems*, 34:20733–20744, 2021.

[47] Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

[48] Adam undefinedcibior, Ohad Kammar, and Zoubin Ghahramani. Functional programming for modular bayesian inference. *Proc. ACM Program. Lang.*, 2(ICFP), jul 2018.

[49] Gail Weiss, Yoav Goldberg, and Eran Yahav. Extracting automata from recurrent neural networks using queries and counterexamples. In *International Conference on Machine Learning*, pages 5247–5256. PMLR, 2018.

[50] Andy Yang and David Chiang. Counting like transformers: Compiling temporal counting logic into softmax transformers. *arXiv preprint arXiv:2404.04393*, 2024.

[51] Fabian Zaiser, Andrzej Murawski, and Chih-Hao Luke Ong. Exact bayesian inference on discrete models via probability generating functions: A probabilistic programming approach. *Advances in Neural Information Processing Systems*, 36, 2024.