Jianlin "Herman" Li

jianlin.li@uwaterloo.ca | jianlin-herman-li.github.io
Google Scholar: BBoZG7gAAAAJ | linkedin.com/in/jianlin-herman-li
Waterloo, Ontario, Canada

Ph.D. in **Computer Science**
specialized in **programming languages** & **compilers**

Objective

I am interested in software safety and efficiency across the entire lifecycle: from **design** (**model checking** and **formal methods**) to **programming** (**program analysis** and **formal verification** via automatic and **interactive theorem proving**) to **compilation** (**type checking**, **optimization**, and **compiler correctness**).

Education

University of Waterloo, Ph.D. in Computer Science — Sep 2021 – **present**
Supervisors: Yizhou Zhang, Ondřej Lhoták — Cumulative GPA: 96.2/100

Saarland University, Saarbrücken, Germany – Exchange (Grade: 1.3, German Grading Scale) — Sep 2019 – Aug 2020
Institute of Software, Chinese Academy of Sciences, M.Sc. in Computer Science — Sep 2018 – Aug 2021
Nanjing University of Aeronautics and Astronautics, B.Sc. in Computer Science — Sep 2013 – Jun 2018

Experience

Research Assistant (Compiler Projects for Probabilistic Programming), University of Waterloo — Sep 2021 – Present

- I design **probabilistic programming languages** (**PPLs**), develop their **semantics**, and implement **compilers** with novel compilation strategies tailored to different inference methods (e.g., variational Bayesian inference, variable elimination) and **compilation artifacts**, enabling efficient, reliable, and certified probabilistic programming:
  - **Geni** (**ICFP 2025**): A compiler implemented in **Rust** that compiles functional-programming–style probabilistic programs into **generating functions**, a compact and exact compilation target representing measures. Proposed the first **design** and **formalization** of GF compilation for a **functional** PPL; implemented a **live variable analysis** optimization; and demonstrated scalability competitive with **binary decision diagram**-based inference.
  - **Mappl** (**PLDI 2024**): A compiler implemented in **OCaml** that enables scalable inference by factorizing recursive probabilistic programs via **continuation-passing-style** (**CPS**) compilation and **information-flow typing** (with a denotational, **logical-relations** model). Compilation artifacts recover expert-designed **polynomial-time** algorithms (e.g., the forward algorithm for **HMM**) without human intervention, whereas baselines require **exponential time**.
  - **Fidelio** (**POPL 2023**): A compiler that embodies symbolic methods (e.g.,**type system**, **program analysis**) aiding **neural-network**–based inference. Consistently improved training and inference performance over mean-field and **LSTM** baselines. (**PyTorch**, **importance sampling**, **variational inference**, **deep amortized inference**).
- Implemented maximum-entropy **reinforcement learning** via variational inference in **Pyro**, a deep Universal PPL.

Teaching Assistant (tutorials, office hours, grading), University of Waterloo — Jan 2022 – Aug 2025

- Computer Architecture, Compiler Construction, Logic and Computation, Foundations of Sequential Programs.

Visiting Researcher, Universität des Saarlandes, Saarbrücken, Germany — Sep 2019 – Aug 2020

- Worked on automated safety **verification** of programs invoking **neural networks** (published at **CAV 2021**):
  - Trained a heterogeneous controller combining neural networks and handcrafted components (C programs) via **reinforcement learning** in an **OpenAI** Minigrid–style environment, enabling autonomous decision making.
  - Integrated **Clam**, a state-of-the-art **LLVM** bitcode static analyzer, with **DeepSymbol** (my neural network verifier), enabling a precise neuro-aware verification pipeline (previously infeasible to verify without our approach).
- Generalized Minsky Machine Halting Problem $\preccurlyeq_m$ 2-Counter Machine Halting Problem in **Rocq** (formerly **Coq**).

Research Assistant (Formal Verification), Institute of Software, Chinese Academy of Sciences — Sep 2018 – Aug 2021

- Developed **DeepSymbol** (**SAS2019**), a **verification** tool for the robustness of **deep neural networks** (**DNNs**).
  - Proposed a novel **symbolic-propagation** method improving **abstract interpretation** for **DNN analysis**.
  - Resolved issues in open-source projects, e.g., memory leak in **Zonotope** of **APRON** numerical abstract domain library that prevented scaling, caused by an ABI mismatch leading to memory layout and calling convention errors.
  - Bring benefits of up to **549%** speedup (9.16h → 1.41h) to a state-of-the-art **SMT**-based **verifier**.

Technologies

Languages and Frameworks: Rust, OCaml, C++, LLVM, Python, Java, PyTorch, Pyro, JAX, Rocq, Coq, Lean 4, Agda.
Skills: Compiler Construction, Type System, Program Analysis, Program Optimization, Formal Verification.

Thesis: *Design and Implement Probabilistic Programming Languages for Sound and Scalable Inference.*

Compilers	Venues	Inference Methods	Compilation Artifacts	Keywords
Pal	Ongoing	Enumeration	JAX Array Programs	Vectorization
Geni[1]	ICFP'25	Knowledge Compilation	Probability Generating Functions	Compiler Optimization
Mappl[2]	PLDI'24	Variable Elimination	CPS-Transformed Factor Functions	Program Partitioning
Fidelio[3]	POPL'23	Variational Inference	Neural-Aided Guide Program	Absolute Continuity

Publications (Probabilistic Programming Languages · Semantics · Compilers · Type Systems)

[1] **Jianlin Li** and Yizhou Zhang.
Compiling with Generating Functions.
Proc. ACM Program. Lang., 9(**ICFP 2025**, Singapore).

[2] **Jianlin Li**, Eric Wang, and Yizhou Zhang.
Compiling Probabilistic Programs for Variable Elimination with Information Flow.
Proc. ACM Program. Lang., 8(**PLDI 2024**, Copenhagen, Denmark).

[3] **Jianlin Li**, Leni Ven, Pengyuan Shi, and Yizhou Zhang.
Type-Preserving, Dependence-Aware Guide Generation for Sound, Effective Amortized Probabilistic Inference.
Proc. ACM Program. Lang., 7(**POPL 2023**, Boston, United States).

Publications (Formal Verification · AI Safety · Adversarial Machine Learning · Model Checking)

[4] Maria Christakis, Hasan Ferit Eniser, Holger Hermanns, Jörg Hoffmann, Yugesh Kothari, **Jianlin Li**, Jorge A. Navas, and Valentin Wüstholz.
Automated safety verification of programs invoking neural networks.
*Computer Aided Verification - 33rd International Conference, **CAV 2021**.*

[5] Pengfei Yang, Renjue Li, **Jianlin Li**, Cheng-Chao Huang, Jingyi Wang, Jun Sun, Bai Xue, and Lijun Zhang. Improving neural network verification through spurious region guided refinement.
*Tools and Algorithms for the Construction and Analysis of Systems - 27th International Conference, **TACAS 2021**, as Part of **ETAPS 2021**, Luxembourg.*

[6] Renjue Li, **Jianlin Li**, Cheng-Chao Huang, Pengfei Yang, Xiaowei Huang, Lijun Zhang, Bai Xue, and Holger Hermanns.
PRODeep: A Platform for Robustness Verification of Deep Neural Networks.
***ESEC/FSE 2020** : 28th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, USA.*

[7] **Jianlin Li**, Jiangchao Liu, Pengfei Yang, Liqian Chen, Xiaowei Huang, and Lijun Zhang. Analyzing Deep Neural Networks with Symbolic Propagation: Towards Higher Precision and Faster Verification.
***SAS 2019**: 26th Static Analysis Symposium, Porto, Portugal.*

[8] Hongfei Fu, Yi Li, and **Jianlin Li**.
Verifying Probabilistic Timed Automata Against Omega-Regular Dense-Time Properties.
***QEST 2018**: 15th International Conference on Quantitative Evaluation of SysTems, China.*

Awards

- Competitive programming:
  - 2015: Silver Medal, **ACM-ICPC** Shanghai Metropolitan Programming Contest
  - 2014: Silver Medal, **ACM-ICPC** Asia Regional Contest, Anshan Site
- Scholarships:
  - 2023: David R. Cheriton Graduate Scholarship (awarded to **5 recipients**)
  - 2021: University of Waterloo Entrance Scholarship
  - 2020, 2014: China National Scholarship (**Top 0.2%**)
  - 2020, 2019: Institute of Software, CAS First-Class Academic Scholarships (**Top 10%**)

Services

- **Subreviewer**: LICS 2018, TASE 2019, FM 2019, FMAC 2019, TACAS 2021, PLDI 2024, OOPSLA 2024.
- **Student Volunteer**: CONCUR 2018, SSFM 2018, SSFM 2019, LICS 2020.