Jianlin "Herman" Li

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

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

Objective

I care about software safety and efficiency across the entire life cycle: from **design** (**model checking**), to **program** (**program analysis**, and **formal verification** via automatic and **interactive theorem proving**), to **compilation** (**type checking**, **optimization**, and **compiler correctness**).

Education (Anticipated graduation date: 04/26)

University of Waterloo, Ph.D. in Computer Science — Sep 2021 – **Expected April 2026**
Supervisors: Yizhou Zhang, Ondřej Lhoták — Cumulative GPA: 96.2/100
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(Probabilistic Programming / Compiler Projects), University of Waterloo — Sep 2021 – Present

- I design **probabilistic programming languages** (**PPLs**) 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** (**PLDI 2025**): A compiler implemented in **Rust** that compiles functional programming–style probabilistic programs into generating functions, a compact and exact compilation target representing measures. Gave the first **design** and **formalization** of GF compilation for a functional PPL;implemented a **live-variable analysis** optimization; and demonstrated scalability competitive with **BDD**-based inference.
  - **MAPPL**(**PLDI 2024**): A compiler implemented in **OCaml** that enables scalable inference by factorizing recursive probabilistic programs via **information-flow typing** (with a denotational, **logical-relations** model). Compilation artifacts recover **polynomial-time** algorithms (e.g., forward algorithm for HMMs, inside–outside algorithm for PCFGs) without human intervention, whereas baselines require **exponential time**.
  - **FIDELIO**(**POPL 2023**): A compiler embodies symbolic methods, e.g., a **type system** and **program analysis**, that aid neural-network–based inference. Consistently improved training/inference over mean-field and **LSTM** guide baselines across benchmarks. (**Pyro PPL**, **PyTorch**, **LSTM**, **Deep Amortized Inference**)
- Implement maximum entropy **reinforcement learning** via variational inference in **Pyro**, a universal probabilistic programming language (PPL) written in Python and supported by **PyTorch** on the backend.

Teaching Assistant(taught tutorials, held office hours, etc.), University of Waterloo — Sep 2021 – Aug 2025

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

Research Assistant & **Exchange Student**, Saarland University (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 (impossible to verify without our approach).
- Generalized Minsky machine halting problem $\preccurlyeq_m$ 2 counter machine halting problem in **Coq**.

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

- Developed **DEEPSYMBOL**, a **verification** tool for the robustness of **deep neural networks** (**DNNs**).
  - Propose a novel **symbolic propagation** method improving **abstract interpretation** for **DNN analysis**.
  - Fixed a severe memory leak in the Zonotope domain of the **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 $\rightarrow$ 1.41h) to a state-of-the-art **SMT**-based **verifier**.

Technologies

Languages/Frameworks: Rust, OCaml, C++, LLVM, Python, Java, PyTorch, JAX, Coq, Lean 4, Agda.
Skills: Compiler Construction, Formal Methods, Formal Verification, Theorem Proving, Program Analysis.

Publications (Probabilistic Programming Languages · 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 InformationFlow.
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 Safty · 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

- 2023: David R. Cheriton Graduate Scholarship
- 2021: University of Waterloo Entrance Scholarship
- 2020: China National Scholarship (Top 0.2%)
- 2020/2019: Institute of Software, CAS First-Class Academic Scholarships (Top 10%)
- 2015: Silver Medal, ACM-ICPC Shanghai Metropolitan Programming Contest
- 2014: Silver Medal, ACM-ICPC Asia Regional Contest, Anshan Site
- 2014: China National Scholarship (Top 0.2%)
- 2014: Winning Prize, RoboCup China Open Soccer Simulation 2D

Services

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