Di Wang

Yanyuan Mansion 520, 151 Zhongguancun N Ave, Haidian District, Beijing 100084
wangdi95@pku.edu.cn • https://stonebuddha.github.io/

Bio

I am an incoming Assistant Professor at Peking University in Fall 2022. I received my Ph.D. from Carnegie Mellon University under the supervision of Prof. Jan Hoffmann. My research focuses are programming languages, quantitative verification, and probabilistic programming; my broader interests include type theory, program synthesis, concurrency, and Bayesian inference. During my Ph.D., I built an effective toolkit for rigorous and automatic analysis of probabilistic programs (PLDI'18, MFPS'19, ICFP'20, PLDI'21), the first coroutine-based paradigm for sound programmable Bayesian inference (PLDI'21), the first sound and relatively complete worst-case input generation algorithm (POPL'19), and the first resource-aware synthesizer for recursive programs (PLDI'19, ICFP'20).

Education

Carnegie Mellon University **Pittsburgh, PA, USA**
Ph.D. in Computer Science *Aug 2017 – May 2022*
Advisor: Prof. Jan Hoffmann
Thesis: *Static Analysis of Probabilistic Programs: An Algebraic Approach*

Peking University **Beijing, China**
Bachelor of Science (with Honors) in Computer Science & Technology *Sep 2013 – Jun 2017*
GPA: 3.83/4.0 (**ranked 3rd** out of ~200)
Advisor: Prof. Yingfei Xiong
Thesis: *Accelerating Program Analyses by Conditional Summarization with Datalog*

Research Experiences

Facebook **Seattle, WA, USA**
Research intern, supervised by Dr. Herman Venter *May 2020 – Aug 2020*
Topics: Formal Verification of Rust Code, Side Channel Analysis of Blockchain Code

Massachusetts Institute of Technology **Boston, MA, USA**
Research intern, supervised by Prof. Adam Chlipala *Sep 2016 – Jan 2017*
Topics: Type System for Complexity Analysis, Complexity Preserved Compiler

University of Wisconsin–Madison **Madison, WI, USA**
Research intern, supervised by Prof. Thomas Reps *Jun 2016 – Aug 2016*
Topics:Probabilistic Reasoning about Side Channel Attacks, Expectation Invariant Analysis of Probabilistic Programs

Peking University **Beijing, China**
Research assistant, supervised by Prof. Lu Zhang and Prof. Yingfei Xiong *Sep 2015 – Jun 2017*
Topics: Complete Library Summarization for Program Analyses, Pointer Analysis for Java

Publications

Refereed Conference Papers

[1] **Di Wang**, Jan Hoffmann, and Thomas Reps. Sound Probabilistic Inference via Guide Types. In *42nd Conference on Programming Language Design and Implementation (PLDI'21)*, 2021.

[2] **Di Wang**, Jan Hoffmann, and Thomas Reps. Central Moment Analysis for Cost Accumulators in Probabilistic Programs. In *42nd Conference on Programming Language Design and Implementation (PLDI'21)*, 2021.

[3] **Di Wang**, David M. Kahn, and Jan Hoffmann. Raising Expectations: Automating Expected Cost Analysis with Types. In *International Conference on Functional Programming (ICFP'20)*, 2020.

[4] Tristan Knoth, **Di Wang**, Adam Reynolds, Jan Hoffmann, and Nadia Polikarpova. Liquid Resource Types. In *International Conference on Functional Programming (ICFP'20)*, 2020.

[5] Tristan Knoth, **Di Wang**, Nadia Polikarpova, and Jan Hoffmann. Resource-Guided Program Synthesis. In *40th Conference on Programming Language Design and Implementation (PLDI'19)*, 2019.

[6] **Di Wang**, Jan Hoffmann, and Thomas Reps. A Denotational Semantics for Low-Level Probabilistic Programs with Nondeterminism. In *Mathematical Foundations of Programming Semantics XXXV (MFPS'19)*, 2019.

[7] **Di Wang** and Jan Hoffmann. Type-Guided Worst-Case Input Generation. In *46th Symposium on Principles of Programming Languages (POPL'19)*, 2019.

[8] **Di Wang**, Jan Hoffmann, and Thomas Reps. PMAF: An Algebraic Framework for Static Analysis of Probabilistic Programs. In *39th Conference on Programming Language Design and Implementation (PLDI'18)*, 2018.

[9] Peng Wang, **Di Wang**, and Adam Chlipala. TiML: A Functional Language for Practical Complexity Analysis with Invariants. In *International Conference on Object-Oriented Programming, Systems, Languages, & Applications (OOPSLA'17)*, 2017.

[10] Hao Tang, **Di Wang**, Yingfei Xiong, Lingming Zhang, Xiaoyin Wang, and Lu Zhang. Conditional Dyck-CFL Reachability Analysis for Complete and Efficient Library Summarization. In *26th European Symposium on Programming (ESOP'17)*, 2017.

Other Publications

[11] Ankush Das, **Di Wang**, and Jan Hoffmann. Probabilistic Resource-Aware Session Types. Working paper, 2021.

[12] **Di Wang**, Jan Hoffmann, and Thomas Reps. Expected-Cost Analysis for Probabilistic Programs and Semantics-Level Adaption of Optional Stopping Theorems. Working paper, 2021.

Teaching and Mentoring Experience

- **Guest Lecturer** – *Foundations of Quantitative Program Analysis*, Carnegie Mellon University 2019
- **Teaching Assistant** – *Bug Catching: Automated Program Verification*, Carnegie Mellon University 2020
- **Teaching Assistant** – *Programming Language Semantics*, Carnegie Mellon University 2019
- **Teaching Assistant** – *Introduction to Computer Systems*, Peking University 2015

- **Mentor** – Vanshika Chowdhary, *Programmable Gibbs sampling with linear types* 2021
- **Mentor** – Mohamed Lotfi, *Synthesis of probabilistic programs that generate handwritten digits* 2021
- **Mentor** – Charles Yuan, *Exact Bayesian inference with distribution transformers* 2019

Professional Activities

- **Artifact Evaluation Committee Member** – POPL'19, POPL'20, CAV'20
- **External Reviewer** – ICALP'18, LICS'19, LICS'20, LICS'21, LICS'22, ESOP'20, ESOP'21, POPL'22, FoSSaCS'22

Scholarships and Awards

- China National Scholarship 2014, 2016
- Huawei Scholarship 2015
- Silver Medal (5th place) in the 39th Annual ACM-ICPC World Finals 2015
- Gold Medal (1st place) in the 39th ACM-ICPC Asia Regionals Anshan site 2014
- Gold Medal (9th place) in the 38th ACM-ICPC Asia Regionals Changchun site 2013

Talks

Conference Presentations

- Sound Probabilistic Inference via Guide Types, *PLDI'21*. Jun 2021
- Central Moment Analysis for Cost Accumulators in Probabilistic Programs, *PLDI'21*. Jun 2021
- Raising Expectations: Automating Expected Cost Analysis with Types, *ICFP'20*. Aug 2020
- Liquid Resource Types, *ICFP'20*. Aug 2020
- A Denotational Semantics for Low-Level Probabilistic Programs with Nondeterminism, *MFPS'19*. Jun 2019
- Type-Guided Worst-Case Input Generation, *POPL'19*. Jan 2019
- PMAF: An Algebraic Framework for Static Analysis of Probabilistic Programs, *PLDI'18*. Jun 2018

Seminar Presentations

- Type-Based Resource-Guided Search, *Peking University*, Programming Language Seminar. Oct 2020
- Taint Analysis for Blockchain Code, *Facebook*, Novice Seminar. Aug 2020
- Automating Expected Cost Analysis with Types, *Facebook*, Novice Seminar. Jun 2020

Projects

Static Tag Analysis of Rust Code

Research Intern at Facebook *May 2020 – Aug 2020*

- Studied the formal semantics of Rust and the static analysis tool MIRAI.
- Proposed and implemented a static tag analysis for Rust; the analysis keeps track of inter-procedural information flow, and allows user to customize tag propagation behavior of primitive operations.
- Applied the static tag analysis to analyze side-channel vulnerabilities of blockchain code.

SIMD Vectorization in In-Memory DBMSs for OLAP Applications

Optimizing Compilers for Modern Architectures, Carnegie Mellon University *Feb 2018 – May 2018*

- Proposed an optimization that uses vectorization in just-in-time query compilation.
- Implemented two approaches that use LLVM to emit SIMD instructions to vectorize predicate evaluation in Peloton, an in-memory DBMS developed by Carnegie Mellon Database Group.
- Achieved a significant speedup (avg. 1.5×) on complex SQL queries.

Predicting the Efficiency of Exact Inference Methods in Bayesian Network

Graduate Artificial Intelligence, Carnegie Mellon University *Apr 2018 – May 2018*

- Reviewed exact inference methods for Bayesian networks from both the statistics and the programming languages community.
- Proposed and implemented a machine-learning-based algorithm that predicts which exact inference method would work best on a given Bayesian network.
- Achieved 72% prediction accuracy on a synthetic test set.