

MIKE (DEYUAN) HE

✉ mikehe@princeton.edu · in @Mike He

EDUCATION

Princeton University, Princeton, NJ 2022 – Est. 2027

Ph.D. in Computer Science

Advisors: Prof. Aarti Gupta

Fields of study: Programming Languages (PL); Compilers; Formal Methods; PL Theory

University of Washington, Seattle, WA 2018 – 2022

B.S. in Computer Science, GPA: 3.89/4.0 (CUM LAUDE)

Advisors: Prof. Zachary Tatlock & Dr. Steven Lyubomirsky

Selected Honor: CRA Outstanding Undergraduate Researcher Award, Honorable Mention (2022)

INDUSTRIAL EXPERIENCE

Taichi Graphics, Remote June. 2022 – Sep. 2022

Compiler R&D Intern (Graphics/Compiler/C++/Python)

Focusing on IR optimizations for [Taichi Language](#), including:

- Refactoring and implementing local matrices for Frontend and CHI IR of Taichi Language
- Extending IR optimizations (e.g. **dead code elimination**) to support the new matrix operations
- Enabling large matrices and optimizations (e.g. **SIMD**) for matrix operations
- Conducting experiments on performance gains; implementing fallback strategies to avoid performance regression on backends that do not support SIMD

Intel Labs, Hillsboro, OR Mar. 2022 – June. 2022

Formal Verification Research Intern (Formal Methods/Python/Dafny)

Developed the **Pyrope** framework for **correct-by-construction** modeling.

- Enabled **proof-driven development** purely in Python
- Encoded the correctness proof of (multi-)montgomery reduction algorithm in Python and verified successfully by compiling to Dafny
- Unified “sources of truth” for correctness proofs and implementations

PUBLICATIONS

- **Mike He**, Haichen Dong, Sharad Malik, and Aarti Gupta. *Improving Term Extraction with Acyclic Constraints*, 2023
- Marisa Kirisame*, Steven Lyubomirsky*, Altan Haan*, Jennifer Brennan, **Mike He**, Jared Roesch, Tianqi Chen, and Zachary Tatlock. *Dynamic Tensor Rematerialization*, 2021

SKILLS

- **Languages:** C/C++, Python, Rust, OCaml, Coq, Dafny, etc. (Open to other languages)
- **Compiler & Applied PL:** Equality Saturation, Static Analysis, Computer-aided Reasoning, SMT
- **PL Theory:** Formal Verification, Type Theory, Mathematical Logic
- **Others:** Computer Graphics, Design and Implementation of Algorithms and Data Structures