MIKE HE

https://ad1024.space

dh63@cs.washington.edu

(206) · 887 · 8588

EDUCATION

University of Washington, Seattle *Sept. 2018—Est. Jun. 2022*

B.S. in Computer Science

- Cumulative GPA: 3.88 / 4
- Field of Studies: Programming Languages & Formal Verification & Compilers & MLSys

PUBLICATIONS

1. Marisa Kirisame*, Steven Lyubomirsky*, Altan Haan*, Jennifer Brennan, **Mike He**, Jared Roesch, Tianqi Chen, and Zachary Tatlock. Dynamic tensor rematerialization, 2021 (*: Equal Contribution)
2. Bo-Yuan Huang*, Steven Lyubomirsky*, Thierry Tambe*, Yi Li, **Mike He**, Gus Smith, Gu-Yeon Wei, Aarti Gupta, Sharad Malik, and Zachary Tatlock. From dsls to accelerator-rich platform implementations: Addressing the mapping gap. In *Workshop on Languages, Tools, and Techniques for Accelerator Design (LATTE'21)*, 2021 (*: Equal Contribution)

SERVICE

→ **MICRO '21**, Artifact Evaluation

→ **CSE 505 - Principles of Programming Languages (Spring 2021)**, Teaching Assistant

EXPERIENCE

3LA, LATTE '21 June. 2020—Now

Research Assistant @ PLSE *Seattle, WA*

· 3LA proposes an end-to-end compilation flow that provides **flexible** and **verifiable** support for custom Deep Learning (**DL**) accelerators. 3LA has a builtin implementation agnostic pattern matching algorithm that is capable of find accelerator supported workloads in DL models leveraging the power of Equality Saturation. Moreover, 3LA addresses the mapping gap between DL models represented in high-level domain-specific languages (DSLs) and target DL accelerators by using Instruction-level Abstraction (**ILA**) as the software-hardware interface. Because ILA models the formal semantics of the target accelerator and can be verified against the RTL implementation, the interface opens up spaces for verification on the correctness of the compiler mapping.

· **Talks & Presentations**:

1. *From DSLs to Accelerator-rich Platform: Addressing the Mapping Gap*, Sept. 2021 at Intel (presented jointly with Steven Lyubomirsky)
2. *Correct & Flexible Support for Custom Accelerators*, Sept. 2021 at SRC ADA Center

Dynamic Tensor Rematerialization, ICLR '21 Oct. 2019—Aug. 2021

Research Assistant @ PLSE *Seattle, WA*

· Dynamic Tensor Rematerialization (**DTR**) is an greedy gradient checkpointing algorithm. DTR **enables** training Deep Learning (**DL**) models on memory-constrained devices (e.g. GPUs, FPGA-based accelerators). Unlike previous approaches, DTR does not need any information of the DL model architectures ahead-of-time; instead it saves memory by evicting and recomputing tensors **on-the-fly**, i.e. trading time for memory, which further exploit opportunities of using gradient checkpointing on DL trainings. DTR is comparably efficient as previous approaches: it requires only $\mathcal{O}(N)$ more forward computations when training a N-layer linear feed-forward neural network with an $\Omega(\sqrt{N})$ memory budget.

Paul G. Allen School, University of Washington Mar. 2021—June. 2021

Teaching Assistant *Seattle, WA*

· Worked as TA for **Principles of Programming Languages** (CSE 505)

· Helped re-designing CSE 505 and developing course materials for various topics about PL and formal verification (**Hoare Logic**, **Lambda Calculus** and **System F**, etc.) in **Coq**.

· Held office hours on weekends and shared tricks used in Coq tactics and Coq programming.

· Coordinate grading of all homework assignments.