Jack Cashman

cashman3@purdue.edu | jackfcashman@gmail.com | +1 (516) 659-2087
cashmanjack.github.io | Google Scholar | LinkedIn | GitHub

Education

University of Rochester **August 2022 - May 2026**
Bachelor of Science, Honors in Computer Science Rochester, NY
Purdue University **August 2026 - Present**
Ph.D. in Electrical and Computer Engineering West Lafayette, IN
Relevant Coursework: Advanced Compiler Optimizations, Machine Learning Systems, Computer Organization, Operating Systems, Programming Language Design, Data Structures & Algorithms, Computer Models & Limitations, Honors Research Problem Seminar, Computer Networks.

Research Experience

University of Rochester **May 2024 – July 2024**
Research Assistant Under Prof. Chen Ding

- **Loop Asymptotic Locality Analysis (LALA):** Developed a symbolic tool to model the reuse distance distribution of affine programs, reducing analysis time from doubly exponential to linear relative to program size.
- **Zipfian Caching for Web Workloads:** Designed and evaluated caching policies to outperform LRU on dynamic Zipfian web workloads, achieving a faster convergence to the optimal hit rate.
- **LLM Text Detection:** Used locality measurements of data movement distance for varying N-gram text representation to classify text as human vs. LLM-generated, building an N-gram trace generator for feature extraction.

University of Utah **May 2025 – July 2025**
Research Assistant Under Prof. Ponnuswamy Sadayappan

- Investigated sparse matrix reordering schema cache hierarchy utilization and it's effects on Sparse Matrix Vector Multiplication (SpMV) performance.
- Reduced error of a machine learning model for pairwise ranking of data layouts based on predicted L1D, L2, and L3 cache misses, addressing the fragility of traditional reordering heuristics.
- Implemented and validated hardware counter frameworks (**VTune, PAPI, LIKWID, Perf**) to capture precise cache performance data for model training and empirical validation.
- Conducted controlled experiments by disabling hardware prefetching and varying compiler optimization levels to isolate performance effects and align empirical results with model predictions.

Honors Thesis **August 2025 – April 2026**
University of Rochester, Dept. of Computer Science

- Independent research project to develop a novel sparse matrix reordering solution. Approach based on directly leveraging analytical cache modeling to drive reordering decisions.
- Designing an analytical cache model that directly targets performance factors including cache capacity, associativity, latency, and replacement policy, prefetchers, thread count, and more. This method of reordering departs from traditional graph theory approaches.
- Integrating analytical prediction with genetic search and heuristics, validating results with hardware counters.
- Parallelizing reordering search and analytical modeling for improved speed of reordering.

Teaching Experience

Teaching Assistant **Aug 2024 – April 2026**
University of Rochester

- TA'ed Computer Organization and Programming Language Design and Implementation
- Led study sessions and office hours to help undergraduate students understand core systems concepts, including memory hierarchy, instruction sets, and digital logic.
- Wrote, graded, and proctored exams.
- Graded student projects and provided feedback on assembly language and C programming assignments.

Selected Projects

Tiny-QMOE **Fall 2024**
Course Research Project, Machine Learning Systems

- Engineered a quantization and dictionary-based compression pipeline to enable inference of large language models (LLMs) on memory-constrained devices (e.g., mobile phones).
- Achieved **23–35x compression** on LLaMA models with minimal accuracy loss on MMLU and ARC benchmarks.
- Gained hands-on experience with GPU memory hierarchy, low-latency compression algorithms, and LLM benchmarking.

Mid-End Compiler Optimization Framework **Spring 2025**
Course Project, Advanced Compilers (Software Analysis & Improvement)

- Built a compiler optimization framework from scratch, operating on an intermediate representation.
- Implemented classic and advanced optimizations including dead code elimination, partial redundancy elimination, and constant propagation using abstract interpretation.

Latency Based Tiling **Spring 2025**
Course Research Project, Honors Undergraduate Problem Seminar

- Built a triangle loop based cache latency micro benchmark for approximating cache capacities for L1D, L2, and L3 caches.
- Leveraged the cache approximation for polyhedral program tiling to produce a 21% performance speedup with effectively no compilation overhead unlike auto-tunning.

Publications

- **Preprint:** Jack Cashman. *Latency Based Tiling.* arXiv preprint arXiv:2510.15912, 2025.
- **Preprint:** Jack Cashman, Jiaqi Nie. *Tiny-QMoE.* arXiv preprint arXiv:2509.22951, 2025.

Technical Skills

Languages: Rust, C, RISC-V Assembly, x86-64 Assembly, Coq, CUDA

Developer Tools: Git, Linux/Unix Shell, VTune, PAPI, LIKWID, Perf, GDB

Areas of Interest: Compiler Optimization, Locality, Cache Modeling, Computer Microarchitecture, Hardware Accelerators, Type Systems/Safety, Machine Verified Proofs