Ziang Li

Cover Letter | ziangli@umich.edu | 734-649-3632 | LinkedIn| GitHub

GPU & LLM performance expert, highly experienced in CUDA C++, LLM E2E performance optimizations, and GPU architecture.

Education

University of Michigan, Ann Arbor Ann Arbor, MI
B.S. in Computer Science, with distinction, GPA 3.91/4 2022/08 - 2024/12

The Chinese University of Hong Kong, Shenzhen Shenzhen, China
Finance, Computer Science & Engineering, 2020 & 2021 Academic Dean's List, transferred to UMich 2020/08 - 2022/06

Coursework: GPU Programming *(A+, 1st on leaderboard)*, Advanced Computer Architecture *(A, 98% in exams)*, Compilers, Parallel Computing, Machine Learning, Computer Vision, Web Systems, Theory of Computation, Financial Accounting.

Experience

Google LLC New York City, NY

- **L4 Software Engineer** - *GPU Performance Team* 2025/08 – Present
  - Joined as L4 for efficient Gemini 2.5 Flash serving on Hopper & Blackwell GPUs, guided software stack roadmap and next-gen model performance projections and codesign, mainly working on Mosaic GPU, JAX, XLA.
  - Improved Gemini Flash Attention kernel by 20% with novel techniques, on top of already optimized production code.

NVIDIA Corporation Santa Clara, CA

- **AI Developer Technology Engineer** - *DevTech Compute Team* 2025/01 – 2025/07
  - End-to-end and low-level performance optimizations for LLMs, custom Flash Attention for multi-modal LLMs and LLM-based low-latency RecSys, low-precision and multi-GPU inference, using CUDA C++ and Nsight extensively.
  - Some open-source libraries and frameworks I work on include vLLM, SGLang, FlashInfer, OpenAI Triton, etc.
  - Extremely optimize CUTLASS and TensorRT-LLM kernels with arch simulators to exploit PTX & SASS scheduling and cycle-level performance opportunities, collaborating with internal arch, compilers, and fast kernel teams.
- **Architect Intern** - *GPU Full-Chip Architecture Team* 2024/05 – 2024/08
  - Cycle accurate GPU full chip simulation for Blackwell & Rubin arch, mainly writing high-performance multi-threaded C++ 17, contributed ~3000 lines of code, broad collaborations with hardware verification, arch, and fast kernel teams.
  - Conducted architecture research for LLM & HPC workloads, with a focus on complicated cross-die memory latency.

Samsung Electronics - *Samsung Austin Research Center* Austin, TX

- **GPU Software Intern** - *OpenCL Core Compute/ML Team* 2024/01 – 2024/04
  - Built an OpenCL kernel libracy for CNNs with C++ metaprogramming from scratch, optimized for AMD's RDNA arch.
  - Analyzed GPU SASS, worked with compiler team to fix performance bugs for loop unrolling and register spilling.

University of Michigan Ann Arbor, MI

- **ML Research Assistant** - *Electrical and Computer Engineering, VLSI-SP Group* 2023/05 – 2023/12
  - Collaborated with Meta Reality Lab, deployed CNNs on Jetson with TnesorRT, researched power efficient deep learning SoC design, including tensor cores, post-training quantization, network on chip. ***(2nd author, publication)***
- **HPC Research Assistant** - *Aerospace Engineering, Advanced Propulsion Concepts Lab* 2023/05 – 2023/08
  - Developed and optimized adaptive mesh refinement (AMReX) fluid dynamics solver using C++, CUDA and OpenMPI.

Projects

Extremely Optimized GPU Kernels 2023/04 - Present

- **Gemini Flash Attention** - Improved performance by 20%, will be open source soon.
- **FlashInfer Multi-Item Scoring** - FlashAttention 2&3 variant for low-latency RecSys, with fine-grained sparsity exploitation, SASS instruction scheduling tunning, 2.25x vs naive masking and 7% faster than causal, upstreamed.
- **OpenCL kernel library** - A kernel library from scratch in OpenCL and C++ template metaprogramming, including GEMM, im2col & Winograd convolution, optimized for AMD's RDNA architecture with JIT compilation, persistent threads, configurable multilevel software pipelining & tiling, achieved 3.2x speedup, 2000+ lines of code.
- **EECS 471 MXNet** - CUDA convolution with a novel tiling strategy, **1st** in class, 4.6x speedup and 42% faster than 2nd.

EECS 470 Processor - Complete micro-architectural RTL design for an out-of-order MIPS R10K style RISC-V CPU from scratch in SystemVerilog, with N-way superscalar & speculative execution, dynamic scheduling, early branch resolution, precise states, branch predictor, load store queue, ~6,000 lines of code, synthesized and tested.

COALDA - LLVM compiler passes for coalescing GPU memory accesses, and frequent path loop invariant code motion.

ML & CV Projects - (ViT) from scratch, MLE, CNN, GAN, DDPM & DDIM diffusion, NeRF, using Numpy and PyTorch.

Search 485++ - Parallel search algorithms in OpenMPI, on a distributed MapReduce framework in multithreaded Python.

Publications

- SteROI-D: System Design and Mapping for Stereo Depth Inference on Regions of Interest, TinyML, 2nd author