Xiteng Yao

xtyao@bu.edu ♦ xtyao66.github.io ♦ LinkedIn ♦ GitHub ♦ Google Scholar

Computer Engineering PhD candidate at BU graduating in May 2027, researching LLM training and inference systems, distributed systems, LLM runtime, and collective communication. Internship experience at Arm and AWS includes LLM runtime development and GPU performance profiling and modeling.

Education

Boston University Boston, MA
PhD in Computer Engineering Expected May 2027
MS in Computer Engineering
Research Focus: LLM Training and Inference Systems, HPC, Distributed Systems, Networking, SmartNICs, Programmable Switches
GPA: 4.0/4.0

Boston University Boston, MA
BS in Computer Engineering, Minor in Computer Science, Concentration in Machine Learning Graduated May 2023
GPA: 3.9/4.0, Magna Cum Laude

Skills

- Coding: C, C++, Python, Verilog, SystemVerilog, CUDA, Bash, MATLAB, Java, C#, JavaScript
- Tools and platforms: vLLM, TensorRT-LLM, SGLang, Triton, Nsight (nsys), ASTRA-sim, GPUDirect RDMA, NVLink, Arm, AWS, Azure, LLVM, GCC, Linux, Docker, Kubernetes, Git, CI/CD, Vivado, Vitis HLS, Agile/Scrum, MIPS, RISC-V, ROS

Experience

Arm Inc. Seattle, WA
Research Scientist Intern May 2026 – Dec. 2026

- Built **one of the first Attention-FFN Disaggregation (AFD)** runtimes for LLM inference, separating attention and feed-forward execution into distinct serving stages.
- Implemented **context-length-aware request routing** and **expert routing** within the AFD control system to address workload and expert imbalance.
- Evaluated on up to **2.4T-parameter models** across 36 GB200 GPUs (NVL72). Achieved up to **67% higher token throughput** and up to **28% lower p99 inter-token latency (ITL)** than vLLM.

Amazon Web Services (AWS) Santa Clara, CA
Applied Scientist Intern Aug. 2025 – Dec. 2025

- Applied and quantified the effects of **LLM inference-serving optimizations** (prefill/decode disaggregation, continuous batching, prefix caching) in LLM serving deployments; tuned **vLLM** runtime parameters to improve deployment performance.
- Built **KernelSight-LM**, a **GPU-kernel-aware profiling and simulation** toolchain combining kernel-latency, communication, and host-overhead models in a discrete-event scheduler to predict end-to-end LLM inference latency and throughput. Evaluated across six model families, with **2.7% median throughput-prediction error** in the target-measured setting; supports HW/SW co-design and design-space exploration. Preprint on arXiv.

Projects

Optimizing LLM Training across Multiple GPUs Sep. 2024 – Present

- Developed a **SmartNIC-based model** for ZeRO training, using **parameter prefetching, buffering, and dynamic scheduling** to overlap GPU/CPU computation with communication and reduce duplicate parameter gathers.
- Hardware-calibrated cycle-accurate simulation projected a **training throughput speedup of over 1.6×** for a 100B-parameter model versus GPU–CPU offload with ordinary NICs; published at ACM ICS 2025 [1].

Programmable Switches for Efficient Distributed LLM Systems Sep. 2024 – Present

- Designed **SmartFuse-LLM**, an FPGA-based switch datapath that fuses collective communication, residual addition, and RMSNorm, eliminating a network round trip and GPU-side normalization. GPU profiling and RTL co-simulation projected **approximately 50.5% lower per-boundary latency**; published at HEART 2026 [3].
- Developed **PoolStateLens** to evaluate **switch-attached memory pools** for LLM training, combining GB200/NVLink-5 measurements, performance modeling, and **ASTRA-sim** to identify communication bottlenecks and capacity–bandwidth tradeoffs; published at IEEE HPEC 2026 [8].
- Extending this work toward **stateful, programmable switch architectures**, implementing FPGA forward/backward datapaths and designing intermediate-state placement and workload-aware offload policies for LLM training and inference.

FPGA-Accelerated Secure Machine Learning with MPC Sep. 2023 – Present

- Designed an **FPGA architecture** for secure ML training with **multi-party computation (MPC)**, coordinating distributed computation and protocol data exchange across participating nodes.
- Integrated **Coyote's RDMA stack** on **AMD FPGA SmartNICs** for CPU-bypass memory transfers, overlapping communication with computation. Layer-level experiments achieved **2.1× convolution-layer speedup** over existing MPC frameworks; the resulting IEEE HPEC 2025 paper received the **Outstanding Student Paper Award** [2].
- Extending this work toward **GPU–FPGA acceleration** using **GPUDirect RDMA** and advanced interconnect technologies, exploring direct data-transfer paths that reduce host-memory copies and CPU involvement to improve communication efficiency and secure-training scalability.

Machine Learning–Guided Compiler Optimization for FPGAs and GPUs Apr. 2022 – Aug. 2023; 2026 – Present

- Built **RNN, Transformer, and GNN pipelines** to analyze HLS program representations and predict LLVM optimization decisions, exploring supervised and reinforcement learning approaches.
- Demonstrated **3×–23× improvements across evaluated HLS metrics**; published at **LLVM-HPC 2022** [4].
- Extended optimization to **Triton/LLVM GPU compilation on NVIDIA Blackwell**, investigating compiler-pass ordering, LLVM transformations, and **RL-guided instruction scheduling** with numerical correctness checks and measured-runtime evaluation; published at IEEE HPEC 2026 [7].

Linux Operating System Implementation Sep. 2021 – Dec. 2021

- Developed a custom multithreading system for Linux, including a thread library, a round-robin scheduler, and thread-local storage (TLS) for efficient parallel task execution.
- Designed and implemented a **custom Linux file system** within the **Virtual File System (VFS)** layer.

Selected Publications

[1] SmartNIC-GPU-CPU Heterogeneous System for Large Machine Learning Model with Software-Hardware Codesign. ACM ICS 2025.

[2] Accelerating Multi-Party Computation Using Heterogeneous Systems. IEEE HPEC 2025. **Outstanding Student Paper Award.**

[3] SmartFuse-LLM: Fusing Computation into Collectives for Efficient Tensor-Parallel LLM Systems. HEART 2026.

[4] Reinforcement Learning Strategies for Compiler Optimization in High Level Synthesis. LLVM-HPC 2022.

[5] Load Imbalance in HPC Applications: Improved Profiling and New Ways to Use Wasted Cycles. IEEE HPEC 2025.

[6] ACBench: A Benchmark Suite for Auditing Retry and Profiling Feedback in LLM-Driven Code Optimization. IEEE HPEC 2026.

[7] Toward RL-Guided GPU Compiler Optimization: Case Studies on NVIDIA Blackwell GPUs. IEEE HPEC 2026.

[8] PoolStateLens: Measurement-Grounded Evaluation of an On-Switch Memory Pool for LLM Training. IEEE HPEC 2026.

Awards & Funding

- **Boston University Presidential Scholarship** (four-year, **half-tuition**) 2019–2023
- **Red Hat Collaboratory** Student Research Project Funding 2022
- BU Undergraduate Research Opportunities Program (UROP) Funding 2021
- BU Summer Training as Research Scholars (STaRS) Housing Stipend 2021

Service

- **Reviewer** for TPDS, FGCS, TRETS, MIT URTC, FCCM, FPT, H2RC, HPEC, HEART
- Co-applicant for NSF Future Computing Research Funding (NSF 25-543)

Teaching Assistantship

Department of Electrical and Computer Engineering, Boston University 2019–2026

Courses: EC311 Digital Logic Design (2 semesters); EC413 Computer Organization (4 semesters); EC440 Operating Systems; EC528 Cloud Computing; EK103 Linear Algebra; EK307 Electrical Circuits.