Yijiahao (Friedrich) Qi

Email: qiyijiahao@gmail.com, yq335@cornell.edu
Github: github.com/Friedrichqi
Google Scholar

ABOUT ME

First-year Ph.D. student in the Computer System Lab, ECE Department, Cornell University. Before joining Cornell, I received my B.S. in Applied Physics from Peking University (EECS Department) in 2026, advised by Prof. Meng Li. During my undergraduate studies, I spent a year as a research intern in the SHARC Lab, ECE Department, Georgia Tech, working with Prof. Cong (Callie) Hao on efficient Mixture-of-Agents (MoA) serving systems.

- **Research Interest:** optimizing large-scale GPU systems for more efficient machine learning systems.
  1. ***Efficient Agentic Systems***: achieving lower serving latency via co-designing scheduling policy and universal memory management. *Related work:* Faster-MoA (DAC'26)
  2. ***Hardware-aware System/Algorithms Design***: designing tailored systems/algorithms on resource-constrained platforms to enable higher performance. *Related work:* CREATE (ASPLOS'26), DySL-VLA (DAC'26)
- **Hands-on Experience:**
  1. **Efficient Embodied Agents:** fault-tolerant embodied AI (CREATE) and robotic-manipulation acceleration (DySL-VLA).
  2. **Algorithm:** PEFT fine-tuning of small models with LLaMA-Factory; customized metrics for dynamic early exit.
  3. **System:** KV-cache management and transmission; prefill–decode scheduling and partitioning in multi-agent serving.
  4. **Hardware:** sparse matrix-matrix multiplication (SpMM) accelerator and systolic-array design in Verilog.

Education

Peking University — Beijing, China
B.S. in Applied Physics; EECS Department; GPA: 3.75/4.0 (Top 18%); — *Sep 2022 - Jul 2026*
Georgia Institute of Technology — Atlanta, USA
Undergraduate Research Intern; Sharc Lab@ECE Department — *Mar 2025 - Feb 2026*
Core Courses Covered:

- Hardware Foundations of Artificial Intelligence (A+)
- Machine Learning for Electronics Information Engineering (A)
- Undergraduate Research Practice (A)
- Chip Design using High-level Programming Language (A)

Technical Skills & Test Scores

Programming language: C++, Python, Pytorch, Verilog
Developer Tools: Vivado, Vitis, Linux, Git, Docker
TOEFL 113(120): Reading 30(30), Listening 28(30), Speaking 27(30), Writing 28(30)
GRE 329(340): Verbal Reasoning 159(170,80%), Quantitative Reasoning 170(170,91%), Analytical Writing 4.0(6.0,63%)

Publications

1. T. Xie*, **Y. Qi***, J. Wen, Z. Wan, Y. Dong, Z. Wang, S. Cai, Y. Liang, T. Jia, Y. Wang, R. Wang, and M. Li, "CREATE: Cross-Layer Resilience Characterization and Optimization for Efficient yet Reliable Embodied AI Systems", accepted by the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2026.
2. Z. Wang*, **Y. Qi***, H. Chen, Z. Wan, "Faster-MoA: Low-Latency Tree-Structured MoA Serving with Early Exit and Agent-Aware Prefill-Decode Overlap", accepted by the Design Automation Conference (DAC), 2026.
3. Z. Yang, **Y. Qi**, T. Xie, B. Yu, S. Liu, and M. Li, "DySL-VLA: Efficient Vision-language-action Model Inference via Dynamic-static Layer-skipping for Robot Manipulation", accepted by the Design Automation Conference (DAC), 2026.

Research Projects

1. **Reliability assessment of LLM-based embodied AI system (*CREATE: Cross-Layer Resilience Characterization and Optimization for Efficient yet Reliable Embodied AI Systems*)**, accepted by ***ASPLOS'26*** , paper link: CREATE
   Duration: Jul 2024 ∼ Mar 2025 in PKU-SEC-Lab with Prof. Meng Li
   Purpose:
   - Assess reliability of modern LLM-based embodied AI system.
   - Propose error detection and correction methodologies at application, system, circuit level.

   What I do:
   - Apply SmoothQuant's fault-injection methodology, identifying `o_proj` and `down_proj` as the most sensitive layers.
   - Design a lightweight AD circuit that protects the planner, improving robustness from 10^{-7} to 10^{-3} BER.
   - Use QuaRot to remap sensitive weight matrices to a more uniform domain, boosting robustness from 10^{-7} to 10^{-5} BER.
   - Integrate an entropy-aware LDO for dynamic voltage scaling, cutting compute energy by 40.6%.

   Experience gained:
   - **Algorithm:** a. LLM quantization b. Fault-tolerance profiling c. Error detection and correction skills.
   - **Hardware:** Hardware architecture for embodied system.
2. **Mixture-of-Agents (MoA) Acceleration (*Faster-MoA: Low-Latency Tree-Structured MoA Serving with Early Exit and Agent-Aware Prefill-Decode Overlap*)**, accepted by ***DAC'26***, paper link: Faster-MoA
   Duration: Mar 2025 ∼ Nov 2025 in GaTech-SHARC-Lab with Prof. Cong (Callie) Hao
   Purpose: Rival server-level large models with small but powerful models through cross-agent communication.
   What I do:
   - Systematically profile the influence of different configuration on the overall performance of MoA system.
   - Introduce a novel tree architecture for local aggregation.
   - Prefilling and Decoding stage pipeline.
   - Apply PEFT via LLaMA-Factory to diversify proposer exploration directions and decrease mutual semantic overlap.

- KV-Cache management for consecutive layers in MoA system.
- Use semantic similarity based early-exit to prune unnecessary inferences (10 × faster) with only ±1% accuracy variation.

Experience gained:

- **Algorithm:** a. PEFT models with PPO via LLaMA-Factory b. Design customized metrics for early-exit
- **System:** a. Construction of multi-agent system b. KV Cache management and transmission
- **Hardware:** Fine-granularity operator-level pipeline in multi-agent system with DP partition on several GPUs.

3. **VLA Model Inference Acceleration (*DySL-VLA: Efficient Vision-Language-Action Model Inference via Dynamic-Static Layer-Skipping for Robot Manipulation*)**, accepted by ***DAC'26***, paper link: DySL-VLA

Duration: Feb 2025 ∼ May 2025 in PKU-SEC-Lab with Prof. Meng Li

Purpose:

- Access inter-layer similarity of VLA model.
- Propose an adaptive layer-skip scheme to bypass different layers based on task requirements and motion significance.

What I do:

- Profile VLA models' robustness under dynamic layer skipping.
- Quantify activation similarity between adjacent layers (mean ∼ 80%)
- Train adapters and skip controllers in two stages, reducing trainable parameters by 85.7× vs. full-parameter finetuning.
- Figure out the trajectory continuity can reflect the importance of the current action.

Experience gained:

- **Algorithm:** Finetuning of the VLA model with customized MLP modules.
- **System:** a. VLA model's layer-wise modification b. Robotic arm familiarization and action generation paradigm.

Course Projects

1. **CircuitNet — Congestion Prediction for VLSI Layouts**, *Machine Learning for Electronics Information Engineering* final project, github link: CircuitNet

Purpose:

- Forecast cell-level routing congestion from post-placement layout images.
- Explore UNet-style architectures and low-bit quantization for accurate and efficient models in EDA tools.

What I do:

- Implement baseline UNet, deeper UNet, UNet+SE, Double-UNet and train on the CircuitNet dataset.
- Benchmark each model with NRMSE, SSIM and EMD.
- Perform ablation studies on depth, attention, normalization and activation choice.
- Run PTQ and QAT experiments of different bit-width on weights *and* activations.
- Analyze decoder- versus encoder-sensitivity to quantization noise.

Experience gained:

- Hands-on practice with VLSI-specific data preprocessing, tiling and augmentation.
- Deep dive into PTQ vs. QAT workflows, including custom bit-width and metric dashboards.
- Insights into the trade-off among model complexity, inference latency and predictive accuracy.

2. **Sparse Matrix-Matrix (SpMM) multiplication acceleration ASIC design**, *Chip Design using High-level Programming Language* final project, github link: SpMM

Purpose: Design an SpMM accelerator to optimize multiplication between an $N \times N$ row-compressed sparse matrix and an $N \times N$ dense matrix.

What I do:

- Implemented the FAN-network within the Reduction Unit (RedUnit) to efficiently compute partial sums of power-of-2 elements.
- Developed a PE core that converts received row-compressed data into a format compatible with RedUnit and performs Sparse Matrix-Vector (SpMV) multiplication, including cross-boundary sum calculations using a Halo-Adder.
- Designed a PE array architecture that transforms received ordinary matrices into row-compressed format and parallelizes SpMV computations into SpMM operations.
- Integrated double input/output buffer (Dbbuf), increasing the overall throughput by allowing simultaneous data transfer and computation.
- Realized output/weight stationary (os/ws) to support finer control while alleviating I/O occupation.

Experience gained:

- Hierarchical hardware design.
- Hardware optimization schemes like os/ws and Dbbuf.
- Absorbing state-of-the-art architecture from newly published papers.