

Yuhao Ge

Mobile: (217)926-3291 | Email: yuhaoge2@illinois.edu | [Homepage](#) | [LinkedIn](#)

Education

University of Illinois at Urbana-Champaign	M.Sc. Computer Science	GPA: 4.0 /4.0	2023.8 - 2025.5
University of Illinois at Urbana-Champaign	B.Sc. Computer Engineering	GPA: 3.95 /4.0	2019.8 - 2023.5
■ Honors: Highest Honors, Bronze Tablet (3%, 2023), Dean's List (2020&2022)			
Zhejiang University	B.Eng. Computer Engineering	GPA: 3.93 /4.0	2019.8 - 2023.5
■ Honors: Zhejiang Provincial Government Scholarship, First-Class Scholarship (3%, 2020&2022)			

Skills

- **Programming:** Python, C/C++, JavaScript, SystemVerilog, Assembly, SQL
- **Frameworks & Tools:** CUDA, PyTorch, TensorFlow, TVM, Triton, PyG, Flask, React, Node.js, Kubernetes, Grafana, Airflow, AWS, GCP, MongoDB, Neo4j, Redis

Work Experience

- Amazon, Annapurna Labs** | *Software Engineer Intern* | *Compiler, Systems, ML, Accelerators* 2024.5 - Present
- [Neuron Compiler](#) | *Optimize deep learning on AWS AI accelerators (Trainium and Inferentia)*
 - Developed infrastructure for automatic kernel generation, compilation, profiling, and visualization with a defined sweep space
 - Collected data for **DMA access pattern** analysis and introduced a **learning-based DMA latency model**
 - Created **first-generation Autotuning** framework from scratch to support **compiler optimization** and **kernel optimization**
 - Used autotuning to optimize the Matrix Multiply **Fusion Pass**, gaining a 5.6% improvement for popular LLMs like Llama
 - Developed **kernel language** for AI accelerators and support the **kernel optimization** with autotuning, achieving a 4.9% HFU improvement for kernels like Matrix Multiply
 - Built support for **multi-process compilation** and **distributed benchmarking**, reducing tuning time by 88.4%
- NFTGo** | *Machine Learning Engineer Intern* | *Backend Team* | *Python, ML* 2023.2 - 2023.6
- [GoPricing](#) | *An NFT pricing service powered by machine learning*
 - Developed a **Regression Model** for NFT pricing using historical transaction data and NFT features
 - Used **MongoDB** and **Redis** to realize efficient data retrieval, and the **FastAPI** web framework to package the API services
 - Streamlined periodic data processing, model training/updating, and monitoring with Apache **Airflow**
 - Deployed and managed the system using **Docker** and **Kubernetes** and monitored through **Grafana**
- TikTok** | *Software Engineer Intern* | *C++, Lua, Game Engine, AR/VR* 2022.5 - 2022.8
- **Amazing Engine** | *TikTok's Next-Generation 3D Game Engine for AR/VR Effects*
 - Collaborated in developing TikTok's **3D Game Engine**, which empowers users to create/use interactive **AR/VR** stickers
 - Implement a query-based animation system **Motion Matching** in C++ for realistic and responsive avatar control
 - Developed an **SDK for Skeleton Retargeting** in C++ and **Lua**, supporting animation adaptation across character models
 - Integrated the cross-functional team's **Text-to-Animation** algorithm into our game engine using the developed SDK
- University of California, Los Angeles** | *Visiting Student Researcher* | *ML, RL, GNN, FPGA, EDA* 2022.6 - 2022.11
- **GNNDSE** | *An automated design space exploration for automatic FPGA accelerator design* | *Advisor: Prof. Jason Cong*
 - Combined **GNN** with an **ML/RL-based Design Space Exploration** to achieve **FPGA Accelerator Design Automation**
 - Developed a learning-based **Cost Model** with **GNN** as a surrogate of the HLS tool for quick and accurate assessment
 - Optimized DSE by deploying heuristic algorithms such as **Genetic Algorithm** and **Simulated Annealing**
 - Used **Reinforcement-Learning** and **Bandits** for automatic algorithm selection, boosting exploration speed by 11%

Projects

- Optimized GPU code generation framework for SParse reguLar Attention** 2023.8 - Present
- Developed SPLAT, an optimized framework for efficient **sparse-MHSA**, targeting moderate sparsity levels
 - Introduced **Affine Compressed Sparse-Row (ACSR)** format for **regular sparsity** patterns in MHSA
 - Engineered advanced **GPU code-generation algorithms** for ACSR, enhancing sparse-MHSA kernel performance
 - Achieved 2.05x and 4.05x speedups over **Triton** and **TVM** kernels with SPLAT implementation
- Remote Car Control System with Real-time 3D Reconstruction** 2023.1 – 2023.5
- Developed a Raspberry Pi robot car with remote control via joysticks, utilizing PID control and STM32 microcontroller
 - Implemented WiFi-based communication for transmission of commands and RGBD images between the car and server
 - Implemented the **SLAM** framework RTAB-Map on the server for real-time **3D reconstruction**, achieving a 10Hz framerate
- Implement A [Game](#) Efficiently on the FPGA Board** 2022.1 - 2022.5
- Ported the game "Doodle Dump" to FPGA with **SystemVerilog**, achieving low power consumption and high efficiency
 - Implemented a SOC with NIOS II in C to manage complex tasks like USB protocol and memory I/O
 - Consumed only 400KB memory, 0.5w power to achieve a 50hz frame rate, won the **Best Design Prize**