Shawn Xiao

☎ 510-365-0141 401 Shady Ave, Pittsburgh, PA yunzhonx@andrew.cmu.edu yzxiao Shawn-yzXiao

Education

Carnegie Mellon University **Aug 2023 – Dec 2024**
Master of Science, Computer Systems *Pittsburgh, PA*

Coursework: Multimodal Machine Learning, Machine Learning Systems, Computer Systems, Distributed Systems, Computer Networks, Software Construction, Cloud Infrastructure and Services

University of California, Berkeley **Jan 2022 – Jan 2023**
Visiting Student at Electrical Engineering and Computer Science *Berkeley, CA*

Coursework: Artificial Intelligence, Data Structures, Computer Architecture, Extended Reality Development

Southern University of Science and Technology **Sep 2019 – July 2023**
Bachelor of Engineering, Computer Engineering *Shenzhen, China*

Coursework: Deep Learning, Object-oriented Programming, Big Data, Mathematical Logic (TAed)

Technical Skills

Languages: Python, Java, Go, C#, C/C++, CUDA, RISC-V, X86 Assembly, MatLab
ML Frameworks & Libraries: TensorFlow, PyTorch, Numpy, scikit-learn, Langchain, TVM
Technologies & Tools: Docker, Kubernetes, Git, Maven, JUnit/Jest
Cloud & Distributed Computing: AWS, Google Cloud, Hadoop, MapReduce, Spark, Kafka
Concepts: LLM, RAG, CI/CD, GPU Architecture, Design Patterns, Human-Computer Interaction, XR/VR

Experience

Berkeley Artificial Intelligence Research **Jun 2022 – Aug 2022**
Research Intern (Software Engineer) — VR/XR, AI, Unity 3D, python, HCI *Berkeley, CA*

- Co-engineered a **VR** Intelligent Tutoring System, leveraging **Unity 3D** and C# to personalize psychomotor skill training, achieving a 32.3% increase in learning gains and optimizing skill transition comfort for users
- Developed Python scripts to assess real-time physical behavior, collaborated on **Machine Learning** algorithms for adaptive learning, and utilized **Git** and **Agile** methodologies in a team of 6 researchers

Language Technology Institute, CMU **Jan 2024 – May 2024**
Student Researcher — Multimodal Large Model, RAG, fine-tuning, PyTorch *Pittsburgh, PA*

- Developing a Multimodal Information Fusion Model to conduct **Retrieval Augmented Generation(RAG)** for **WebQA** dataset, which contains more than 40,000 snippets and images QA samples.
- Improving QA quality by focusing on model's cross-modal information transfer during reasoning (QA), fine-tuning **large models** to answer queries that require **multi-hop reasoning**.

Projects

ChatYTB: A YouTube chatbot powered by **LLMs** and Vector Database | *AWS, RAG, Full-Stack* **Individual Project**

- Designed and implemented a **full-stack** web application, deployed on **AWS** to interact with YouTube video content
- Deployed **HuggingFace**'s MiniLM model for semantic text embeddings and OpenAI's GPT 3.5 for query retrieval and summarization. Integrated Python's **Flask** for back-end development, used **LangChain** for transcript splitting
- Developed a user-friendly front-end with **HTML/CSS** and **jQuery**, hosted on **NGINX** web server

Tensor Program Optimization for ML Compilation | *TVM, CUDA, GPU* **Machine Learning Systems**

- Developed an optimization pipeline for GeMM + ReLU + add operator using **TVM**, targeting NVIDIA **GPUs**
- Employed **shared memory tiling**, **register tiling**, and **cooperative fetching**, reducing memory accesses.
- Enhanced performance on NVIDIA T4 GPU by integrating **auto-tuning**, reduced latency(**<20ms**) significantly.

Automatic Differentiation System | *Python, NumPy, AutoDiff* **Machine Learning Systems**

- Engineered a Tensorflow-styled autograd framework based on static computational graph, enabling **automatic differentiation**, facilitating gradient computation essential for model training.
- Implemented logistic regression model and trained it on a handwritten digit dataset, achieving **96%** accuracy

Distributed Bitcoin Miner | *Go, UDP/TCP, BaaS, Load Balancer* **Distributed Systems**

- Built a scalable **networking** middleware in **Go** for client-server communication
- Implemented a **UDP**-based protocol that incorporates selected **TCP** features such as sliding windows, timeout-based retransmissions, and heartbeat mechanisms to achieve reliable communication
- Integrated a weighted priority queues **load balancer**, minimized **30%** request/response times for worker nodes