Shouyi Li

sl5632@columbia.edu | +1 (323)-633-8029 | linkedin.com/in/shouyili

EDUCATION

Columbia University New York, NY

- MS in Computer Science (Machine Learning Track), GPA: 4.0/4.0 Aug 2024 – Dec 2025

University of Southern California Los Angeles, CA

- BS in Computer Engineering and Computer Science, GPA: 3.9/4.0 Aug 2020 – May 2024

SKILLS

Languages: Python, C++/C, Java, HTML & CSS, Javascript/Typescript, SQL

ML Stacks: Pytorch, Tensorflow, CUDA, Computer Vision, NLP, Data Mining, Parallel Computing, Distributed Systems, scikit-learn, TensorRT, Triton Inference Server, CUTLASS, GPU computing

Frameworks/Libraries: Spring/Spring-boot, REST API, FastAPI, React, Flask

Cloud/Databases: AWS (EC2, S3, Lambda, Step Functions, RDS, DynamoDB, VPC, EKS, ECR, CloudFormation, CloudWatch, IAM), GCP, MySQL, NoSQL, PostgreSQL, Google Firebase, MongoDB

Tools: Linux, UNIX, Vim, Git, CI/CD, Docker, Kubernetes, Helm, Bash/Shell, VS Code, DataGrip, MapReduce

PROFESSIONAL EXPERIENCES

Amazon Web Services (AWS) Seattle, WA

Software Engineering Intern May 2025 – Aug 2025

- Led the design and end-to-end implementation of a GitOps solution using ArgoCD to automate Helm Charts deployments on Kubernetes clusters, reducing deployment time by 45% across 60+ LLM inference EKS clusters
- Containerized a custom AWS S3 plugin using Docker for ArgoCD to securely retrieve cluster-specific configurations
- Engineered automated synchronization to reduce manual deployment interventions by at least 90%
- Integrated rollback mechanisms and a canary update strategy for safe, progressive delivery
- Collaborated with cross-functional teams to integrate solutions with a wide range of AWS services, including EC2, S3, EKS, ECR, VPC, IAM, Lambda, Step Functions, API Gateway, Route 53, CloudWatch, and CloudFormation
- Utilized Git and Infrastructure as Code (IaC) principles to version control all configurations and CI/CD

Meituan Beijing, China

Software Engineer (AI/ML) Intern May 2024 – Aug 2024

- Optimized LLM inference time to improve prompt response speed by 23%, significantly enhanced API performance for large language model applications such as customer service chatbots and AI agent development platform
- Implemented batch inference optimization for large-scale LLM deployment based on research literatures, used dynamic pruning techniques with a parallel-processed predictor to reduce MLP computations by up to 50%
- Coded customized CUDA compute kernels, reducing GPU memory usage by 26% during inference
- Collaborated with cross-functional teams for end-to-end deployment and facilitating downstream communications

University of Southern California, High Performance Computing Lab, CSI Cancer Lab Los Angeles, CA

Student Researcher Mar 2023 – Aug 2024

- Engaged in project “*Accelerating ViT Inference on FPGA through Static and Dynamic Pruning*” (FCCM 2024)
- Incorporated block pruning and token dropping techniques to accelerate Vision Transformer model inference
- Designed task-specific FPGA accelerator to address inherent computational challenges in ViTs
- Reduced computation complexity by 3.4× with ≈ 3% accuracy drop and a model compression ratio of 1.6×
- Attained 12.8×, 3.2×, 2.1× speedup compared with state-of-the-art implementation on CPU, GPU, and FPGA
- Conducted “*Investigating the Efficiency and Performance Gains of FPGA-accelerated CNN*” (CONF-MLA 2023)
- Developed project “*Deep Morphological Profiling of Immune Cells in Peripheral Blood*”

Sunwood Ecological Engineering Irvine, CA

Software Engineering Intern May 2023 – Aug 2023

- Engaged in an internal system development to help workers manage ecological projects and track progress
- Developed a responsive React frontend, leveraging useMemo and useCallback hooks to optimize rendering and improve application performance, reducing unnecessary re-renders by 40%
- Implemented REST APIs using Spring Boot, incorporating Spring Security to handle authentication and authorization, ensuring secure data access across the platform
- Optimized PostgreSQL database queries by refining query structures, improved data retrieval speeds by 30%
- Containerized and deployed applications using Docker, managed deployment and scaling on AWS EC2 instances