Yinfei Wang

(646) 533 4174 | stevenyfwang1019@gmail.com | github.com/Steven-wyf | linkedin.com/in/ywsteven

EDUCATION

New York University, New York, NY ***GPA:4.0*** Master of Engineering, Computer Engineering **Sep.** 2024 - **Dec.** 2025
University of California, Irvine, CA Bachelor of Science in Computer Science and Engineering **Sep.** 2016 - **Mar.** 2021

PROFESSIONAL EXPERIENCE

USWOO REALTY LLC

Software Engineer Intern

- ***Project Lead for LLaRA++: Cold-Start Music Recommendation System*** **May. 2025 - Sep. 2025**
  - Designed and deployed a **FastAPI**-based inference system (BERT-> Matrix Factorization MLP LLaRA) with Docker, Kubernetes, and MLflow, enabling real-time music recommendation with < 200 ms latency.
  - Built **end-to-end MLOps pipelines** using **Terraform**, **ArgoCD**, and **GitHub Actions**, automating model retraining, canary rollout, and experiment tracking across staging and production environments.
  - Optimized model serving via **ONNX quantization** and **Prometheus-Grafana monitoring**, improving throughput by 45% and ensuring scalable, observable deployments.

USWOO REALTY LLC

Software Engineer Intern

- ***Project Lead for CAG: Cache-Augmented Generation*** **Oct. 2025 - Apr. 2025**
  - Engineered a high-performance inference framework replacing **RAG** retrieval with **KV-cache** optimization, reducing LLM response latency by 40% while improving throughput and memory efficiency.
  - Designed and implemented a **modular Python architecture** with configurable pipelines for model serving, evaluation, and caching, integrating **Docker, argparse CLI**, and dynamic environment loading.
  - Built automated **benchmarking workflows** to evaluate accuracy − latency tradeoffs across datasets (SQuAD, HotpotQA) and deployed distributed testing environments on **CPU/GPU clusters** for scalable experiments.

Huawei Cloud Computing Technology Co. Ltd.

Sr. Software Development Engineer

- ***Project Lead for a Cloud Scalable Non Relational Database Design, Optimization and End-to-End delivery*** **Jan. 2024 - May. 2025**
  - Led the **design and optimization of a distributed non-relational database** built on **MongoDB and RocksDB**, implementing **C+ routing layer enhancements** and **LSMtree based aggregation**, achieving **100k+doc/sec throughput** and millisecond-level latency under high concurrency.
  - Developed **atomic bulk transaction mechanisms** across gateway, forwarding, and storage layers;standardized **RESTful** and **SDK** interfaces to support diverse enterprise use cases such as **multiplayer trading and large-scale payroll processing**.
  - Built a **DevOps-driven CI/CD pipeline** using **Kubernetes**, **Jenkins**, and **Docker**, integrating **Python regression tests**, **Prometheus health monitoring**, and **Shell-based backup automation**, enabling daily incremental releases and improving deployment velocity by 60%.
- ***Project Lead for Intelligent Performance Monitoring and Optimization for KVS Service*** **July. 2023 - Jan. 2024**
  - Designed and implemented an **intelligent performance monitoring platform** for the **KVS service**, embedding **FTDS metric** tracking into **C++ core modules** to visualize latency, throughput, and concurrency across **MongoDB − RocksDB** data layers.
  - Built a **custom metric visualization and benchmarking framework** using **Python**, **Prometheus**, and **Grafana**, supporting user-defined parameters (KV size, concurrency, partition count) and increasing performance analysis efficiency by **95%**.
  - Optimized **high-concurrency I/O and request aggregation** through **RocksDB batch interfaces** and caching mechanisms, reducing **99th percentile latency by 90%** and boosting overall throughput by **86%** under production workloads.
- ***Project Lead for a disaster recovery tool to restore the storage system state based on system log data*** **Feb. 2022 - July.**2023
  - Designed and developed a **disaster recovery framework** for distributed storage systems, building a **REPL- and GFlag-based interactive interface** to extract, decode, and back up system logs after service crashes.
  - Implemented a **log decoder and recovery engine** in C++ to parse persistent WAL logs, locate data corruption at the bit level, and reconstruct **LSM-tree** states to ensure full data consistency up to the point of failure.
  - Authored **user documentation and recovery playbooks** for multiple disaster scenarios, streamlining developer onboarding and improving recovery success rate during system simulations.
- ***Delivered multiple key features like system snapshot on a distributed file system based on AWS-S3*** **July. 2021 - Jan. 2022**
  - Delivered multiple core features for a **distributed file system** built on **AWS S3**, including **automatic file recycling**, **duplication detection**, and **system snapshots** to enhance storage efficiency and data reliability.
  - Developed a **TCC-based** (Try-Confirm-Cancel) **deduplication framework** in **C++ and Python**, automating periodic scans through a **client-worker task queue** for real-time cleanup of redundant or junk files across distributed nodes.
  - Designed and implemented a **configurable snapshot mechanism** supporting user-defined directories and frequencies, coupled with **exponential-backoff retry logic** to ensure consistency and fault-tolerant recovery in large-scale environments.

TECHNICAL SKILLS

Languages : Python, C++, C, Java, Javascript, Go, Rust, TypeScript, HTML, SHELL, CSS, MATLAB, LaTex, LISP, Prolog, JavaFX
Frameworks: ArgoCD, MLFlow, LangChain, React js, Angular, Node.js, Django, Spring Boot, FastAPI, PyTorch, Flask, Vue js
Database : MySql, MongoDB, DynamoDB, RocksDB, Amazon S3, MiniIO, Snowflake, Nginx, LSM tree, Redis
Developer Tools: AWS, Azure, Git, Linux, Nginx, Docker, Kubernetes, jenkins, CI/CD, Prometheus, Kafka, Grafana, RESTful and SDK