Xinle (Eric) Song

310-869-9560 | ericsong311@ucla.edu | linkedin.com/in/xinle-song | github.com/EricSongXinLe

Education

University of California, Los Angeles — Los Angeles, CA
Bachelor of Science in Computer Engineering — *Sep. 2023 - June 2027*

- GPA: 3.9, Dean's Honors List
- Selected Coursework: Parallel Computing, Computer Architecture, Data Structures and Algorithms
- Received the John DeGroff Haller Memorial Scholarship (2025), and the John Richard Leffler Scholarship (2023)

Technical Skills

Languages: C/C++, Python, JavaScript (Node.js/React), Verilog
Parallel Programming & Optimization:OpenMP, MPI, CUDA, SIMD (AVX2/AVX512), loop tiling, cache profiling
Tools & Frameworks: Linux, GDB, Git, perf, Docker, vLLM, Bash, High-Level Synthesis (HLS)

Experience

Software Engineer Intern — Aug. 2025 – Sep. 2025
Tencent — *Shanghai, China*

- Adapted and deployed **Shennong Agentic LLMs** on **NVIDIA H20/A10 GPUs**, completing accuracy validation and performance benchmarking for around 10 models.
- Developed a reusable **Python automation script** for end-to-end testing (data prep, model loading, output collection, report generation), reducing runtime from **7–8 hours to 30 minutes**.
- Developed vLLM modules for **weight decryption** and **license authentication**, ensuring correct model decryption and enforcing license checks for secure delivery.

Undergraduate Research Assistant — Jan. 2025 – Present
UCLA ORCAS Lab

- Added **Sparse Matrix support** to a simulated Tensor Core on the Vortex GPGPU using 2:4 structured sparsity.
- Redesigned **Fused Elementwise Dot Product** (FEDP) to support sparse A, skipping half the FMA operations.
- Designed compact loaders for **Sparse Matrix A** (values and masks), **doubling** the throughput.
- Built tests to verify correctness by computing a reference **D matrix**, optionally enabled with a define flag.

Projects

Parallel Matrix Multiplication Accelerator | *C++, OpenMP, AVX2/512, perf, Linux* — Apr. 2025 – Jun. 2025

- Optimized **GEMM** for large matrices (4096*4096) using **OpenMP** and **SIMD** on x86 and ARM platforms.
- Achieved up to **633x speedup** by applying **loop reordering, tiling**, and AVX-based **FMA micro-kernel**.
- Identified cache miss bottlenecks using **perf**, tuned tiling for **L2** reuse, and enabled **AVX512** for GEMM.
- Reached **133 GFLOPS** on AWS Xeon CPU and **300 GFLOPS** on M1 MacBook, showcasing scalability.

Pipelined RISC-V CPU Simulator | *C++, Processor Design, GNU Make, Git* — Jan. 2025 – Mar. 2025

- Built a **5-stage RISC-V** CPU simulator supporting instruction decode, execute, and out-of-order execution.
- Implemented a **GSharePlus** predictor combining global/local history with dynamic selection to boost accuracy.
- Implemented **RAT** to eliminate data hazards and optimized **RS, ROB, and CDB** for efficient scheduling.

Brewin Interpreter | *Python, Abstract Syntax Tree (AST), Static Typing, Git* — Oct. 2024 – Dec. 2024

- Developed an interpreter for **Brewin**, supporting **static typing**, **user-defined structs**, and **coercion**.
- Added **nested structs**, **default returns**, **value/object reference** passing, and **exception handling.**
- Implemented **static scoping** and **function calls**, optimizing variable lookup with deque and dictionary (**+30%**).
- Achieved **100%** unit test pass rate by verifying all edge cases against language specifications.

Find Your Clubs | *JavaScript, React, Node.js, MongoDB, Git, Tencent Cloud* — Apr. 2024 – Jun. 2024

- Led a team of **6** to build UCLA's first smart club search platform, deployed on **Tencent Cloud VPS** (2C2G).
- Built **RESTful APIs** for **authentication**, **search**, and **recommendation**, with role-based access control.
- Developed reusable React components (e.g., **ClubBlock**) with state control, improving load speed by **20%**.
- Implemented **MD5 password encryption**, input validation logic, and MongoDB-based user authentication.
- Completed the full **DevOps pipeline**, including deployment, debugging, monitoring, and cloud configuration.