

# Shengwei Liu

New York, NY | 646-217-5499 | sl3597@cornell.edu  
<https://sl3597.github.io/>

## EDUCATION

Cornell Tech, Cornell University	New York, USA
MEng in Electrical and Computer Engineering, GPA: 4.019/4.0, Merit Scholarship	May 2026
Relevant Coursework: ASIC, Computer Architecture, Machine Learning, ML Hardware & Systems, Memory-Centric Computing	
University College London (UCL)	London, UK
BEng in Electronic and Electrical Engineering, First-Class Honors (UK Highest Classification)	June 2025

## TECHNICAL SKILLS

- Programming:** Python, C, C#, C++, CUDA, Embedded C, SystemVerilog, MATLAB
- Frameworks & Libraries:** TensorFlow, PyTorch, Keras, scikit-learn, SciPy, NumPy, Pandas, OpenAI Gym, TensorFlow Lite
- EDA Tools:** Synopsys (Design Compiler, PrimeTime), Cadence (Virtuoso, Innovus), Siemens EDA (QuestaSim, Calibre)
- Operating Systems:** macOS, Linux, Windows

## EXPERIENCE

Cornell University, Computer System Lab, NY, USA	Dec. 2025 – Present
Research Assistant	Advisor: Prof. Mohamed Abdelfattah
<ul style="list-style-type: none"><li>Investigated hardware-software co-designed LLM inference acceleration, focusing on low-bit quantization and numeric formats (FP4/MXFP4/NVFP4), decode-stage bottlenecks, and memory bandwidth efficiency.</li><li>Analyzed architectural design choices for ASIC-style compute pipelines and PIM-based memory systems for low-bit inference workloads, identifying scalability limits under memory-bound conditions.</li><li>Analyzed KV-cache quantization framework, validating its memory-accuracy trade-offs and integrating it with LLaMA-3.1 models for efficient long-context inference.</li></ul>	
Tsinghua University, HAS Lab, Beijing, China	May 2025 – Sept. 2025

## PROJECTS

GPT-2 Inference Kernel Optimization (C)	Nov. 2025 - Dec. 2025
<ul style="list-style-type: none"><li>Implemented GPT-2 inference and performed microarchitectural profiling using Cachegrind, gprof, and gcov.</li><li>Optimized the linear layer kernel using SIMD (AVX2/FMA), tiling, loop unrolling, memory flattening, and data-layout restructuring, achieving 3.95× speedup and 10× reduction in instruction count.</li><li>Improved cache locality and memory bandwidth utilization by eliminating double-pointer indirection and applying data-reuse strategies.</li></ul>	
Lightweight Pipelined Encryption Engine (SystemVerilog)	Oct. 2025 – Dec. 2025
<ul style="list-style-type: none"><li>Designed a fully pipelined Simon32/64 lightweight block-cipher engine, exploring 1/2/4/8/16/32-stage microarchitectures to optimize throughput under strict PPA (power, performance, and area) constraints.</li><li>Implemented complete RTL, performed functional verification across behavioral, post-synthesis, and post-layout netlists (QuestaSim), and achieved functional correctness over 1,000 randomized test vectors.</li><li>Completed full ASIC flow including synthesis (Synopsys DC), automatic place-and-route with clock-tree synthesis (Cadence Innovus), and timing closure using post-route STA (PrimeTime).</li><li>Generated final GDSII, resolved DRC/LVS issues (Calibre), and analyzed post-layout performance and power consumption.</li></ul>	
Optimizing ML Kernel Operations (C)	Aug. 2025 – Oct. 2025
<ul style="list-style-type: none"><li>Built core ML kernels (MatMul, Conv, ReLU) from scratch to understand computational bottlenecks.</li><li>Applied advanced optimization techniques including tiling, blocking, and mixed-precision arithmetic to enhance performance.</li><li>Explored multi-threading and parallel execution to improve CPU utilization.</li><li>Integrated sparsity-aware methods to accelerate sparse matrix operations in ML inference.</li><li>Combined all optimized kernels into an end-to-end neural network inference benchmark pipeline and evaluated performance.</li></ul>	
Hand Motion Detection using IMU and PPG Sensors (Python, Embedded C)	Oct. 2024 – Apr. 2025
<ul style="list-style-type: none"><li>Developed a real-time hand motion detection system integrating 6-axis Inertial Measurement Unit and 4-channel Photoplethysmography sensors, enabling Bluetooth-based multi-modal data streaming and edge-side preprocessing.</li><li>Collected 1,800 samples across 6 hand motions, each with 15-time steps and 10 fused features, and utilized SciPy, Keras, and Scikit-learn for preprocessing and feature extraction, including Min-Max normalization, signal filtering, and stratified splitting.</li><li>Trained a stacked LSTM network (25-15-10 units) with dropout and batch normalization via TensorFlow, achieving 94.05% test accuracy and 0.94 macro F1-score, outperforming SVM, KNN, and Random Forest baselines.</li><li>Deployed quantized LSTM models on an Arduino Nano 33 IoT using the LiteRT framework and implemented a Python-based BLE inference pipeline with the Bleak package for real-time motion feedback to a PC without data loss.</li></ul>	

## PUBLICATION

Shengwei Liu, et al. “SHMemora: Protective Key–Value Store on Distributed Shared Memory”. In: *Proceedings of the 42nd IEEE International Conference on Data Engineering (ICDE’26)*, May 2026