Chia-En (Allen) Lu

+1-331-341-8507 | chiaen119@gmail.com | GitHub | LinkedIn

Education

University of California, San Diego Sept. 2024 – Dec. 2025
Master of Science in Computer Science *CA, USA*

National Tsing Hua University Sept. 2019 – Jun. 2023
Bachelor of Science in Computer Science *Hsinchu, Taiwan*

Work Experience

Ambient.AI Feb. 2026 – Present
Software Engineer (ML Infrastructure) *CA, USA*

Model Optimization

- Enabled TensorRT export of a YOLO license plate recognition model by replacing branching stream-selection logic with mask-based batch processing, eliminating dynamic control flow.
- Optimized the YOLO-based pose estimation model inference time 34–40% via TensorRT compilation and cut postprocessing 35–45% by vectorizing NMS across the full batch.
- Eliminated per-call tensor copy overhead via DLPack zero-copy sharing, cutting transfer time from 111.9 ms to 103.7 ms.
- Addressed a 24× inference regression (17 ms → 410 ms) to 11 per-forward graph breaks blocking full-graph compilation
- Corrected a 480× model compilation regression (25 ms → ∼12 s) in a Vision Transformer token-merging integration
- Traced 45,000+ redundant kernel launches per batch via Nsight profiling to a global flag contaminating the shared CUDA context, reducing kernel count by 500×.

Inference Pipeline Optimization

- Reduced cloud inference costs by 80% by implementing scale-to-zero autoscaling for idle model replicas.
- Migrated an 11-model real-time CV inference pipeline (ViT, VLM, YOLO, 7 TRT CNNs) from monolithic PyTorch to NVIDIA Triton, enabling independent GPU memory management and concurrent model execution.
- Cut tensor transport latency 18× (∼55 ms → ∼3 ms) by implementing shared-memory transport between the inference server and production appliance.
- Lowered multi-model inference latency 33% (231 ms → 155 ms) by designing a pipeline-parallel inference scheduler to overlap model preprocessing and GPU execution.
- Profiled the production stack and established severe GPU under-utilization (SM Issue ≤17.2%), confirming CPU scheduling as the bottleneck over GPU compute.
- Built a production observability stack (Redis→Prometheus→Grafana) exposing per-model P50/P95/P99 latencies.

GenseeAI Inc. June 2025 – Sept. 2025
Research Intern (AI Systems & Infrastructure)

- Architected a sandboxed AI agent execution environment on Docker and AWS ECS/ECR with CI/CD and pre-built image caching, reducing container startup latency by 40%.
- Co-developed the full-stack AI agent serving platform supporting multi-turn agentic sessions with tool execution, state management, and per-session usage tracking.
- Designed secure multi-environment APIs with FastAPI and JWT/OAuth for reliable agent session orchestration across development and production.

Taiwan AI Labs June 2024 – Aug. 2024
Software Engineer Intern (Algorithm Team)

- Improved Truku language translation quality 14% (BLEU 28 → 32) and cut training time 13% by fine-tuning a 65M-parameter Transformer with Dynamic Data Selection.
- Achieved BLEU 37.5 and TER 41 on medical domain translation by applying retrieval-augmented data augmentation to a clinical LLM.

Research Experience

Shang Data Lab, UCSD — Mar. 2025 – Feb. 2026

Graduate Researcher, advisor: Jingbo Shang

- Engineered an interactive RL training environment in PyTorch for LLM tool-use, supporting multi-turn reasoning trajectories with tool-call verification and reward shaping.
- Accelerated distributed training 4× via DeepSpeed ZeRO-3 and achieved 12× inference throughput improvement by integrating vLLM, SGLang, and FlashInfer.

Swartz Center for Computational Neuroscience, UCSD — Sept. 2024 – Feb. 2026

Graduate Researcher, advisor: Tzyy-Ping Jung

- Developing a unified cross-spectral estimator for multichannel EEG that jointly fits spectral decomposition, cross-spectral density, and spatial dispersion law in a single model, extracting wave physics from off-diagonal electrode relationships that per-channel analysis discards.

Publications

Kulkarni, A.*; **Lu, C. E.***; Mekala, D.; Srinivasa, J.; Liu, G.; Shang, J. (2026). **TIER: Trajectory-Invariant Execution Rewards for Multi-Step Tool Composition**. *Under review, NeurIPS 2026* (*equal contribution). [Link]

Selected Projects

Matrix Multiplication Optimization — Sept. 2024 – Dec. 2024

C++, CUDA, ARM SVE SIMD

- **CPU (AWS Graviton3):** Achieved ∼20.5 GFLOPS (∼14.6× over naive, exceeding CPU BLAS) by implementing a cache-optimized SIMD microkernel with matrix packing and software prefetching.
- **CUDA (NVIDIA T4):** Reached 3854 GFLOPS at N=2048 — 9.25× over naive and 82% of cuBLAS — by implementing two-level shared memory and register tiling, operating at 88% of theoretical memory bandwidth.
- Diagnosed performance dips at power-of-2 matrix sizes via cache profiling, identifying cache associativity conflicts as root cause; resolved via tile parameter grid search.

End-to-End LLM Systems Design — Jan. 2025 – Mar. 2025

PyTorch, OpenAI Triton, MPI4py

- Developed a from-scratch autodiff engine and trained a single-layer Transformer on MNIST; implemented fused MatMul+LayerNorm and MatMul+Softmax operators with correct backpropagation.
- Achieved 1.41× speedup over PyTorch (0.73 ms vs. 1.02 ms, 2048×2048 on T4) by implementing a fused MatMul+ReLU+Add Triton kernel that combines 3 operations into 1 kernel launch.
- Implemented custom MPI collectives from scratch and built hybrid tensor-parallel + data-parallel MoE; expert-parallel MoE ran 20× faster than tensor-parallel (0.18 ms vs. 3.61 ms per batch).
- Deployed speculative decoding achieving up to 2.3× speedup (56% latency reduction) with 85–92% draft token acceptance under greedy decoding.

Skills

Programming: Python, C++, C, CUDA, Go, JavaScript

ML & Inference: PyTorch, TensorRT, NVIDIA Triton, vLLM, SGLang, FlashInfer, NumPy, OpenCV

GPU & Systems: NVIDIA Nsight Systems, NVIDIA MPS, OpenAI Triton, distributed training, MPI4py, DeepSpeed

Infra & DevOps: AWS, Docker, Kubernetes, Redis, Prometheus, Grafana, Git, GitHub Actions, FastAPI