Aaron Chu

San Jose, CA | 669.946.5814 | aaronchu022@gmail.com | linkedin.com/in/achu02 | github.com/aarochu

Education

Rutgers University — New Brunswick, NJ

BS Computer Engineering — *May 2028*

- **Coursework**: **Distributed Deep Learning**, Probability & Random Processes, Computer Architecture, Data Structures, Software Engineering, Discrete Mathematics

Experience

Software Engineer Intern — June 2026 – Aug. 2026

Magnefy — *Palo Alto, CA*

- Trained and benchmarked **XGBoost**, logistic regression, and **multilayer perceptron** classifiers that tell 3 partial-discharge types apart on 27K sensor pulses, reaching 0.991 macro F1 under grouped 5-fold cross-validation
- Replaced an **ML** anomaly filter with a rule-based screen on raw samples after evaluating both against labeled data; it caught all 695 labeled arcing events at a 1.48% false-positive rate and now runs in production
- Automated processing of sensor data from 9 industrial transformers (598 captures, 193K waveforms) with an **AWS S3→SQS** data pipeline, plus a **Streamlit** tool where operators review, label, and correct results before publishing

Lead Systems Integration Engineer — Sept. 2025 – Present

University Nanosatellite Program — *New Brunswick, NJ*

- Led a 10-member team and built a thread-safe **Python** package polling 4 **I2C** sensors on a **Raspberry Pi**, with per-sensor error tracking and data-age checks so stale readings never reach the flight software
- Authored 3 Critical Design Review test procedures for SPICEsat, a 6-unit **CubeSat**, and presented them to a panel from the Air Force Research Laboratory, Space Dynamics Laboratory, and industry partners

Drone Support Intern — June 2025 – Aug. 2025

Integem — *Cupertino, CA*

- Fused **AprilTag** camera headings with the IMU to cancel yaw drift for autonomous flight on an **ESP32-S3** drone, running 3 **PID** loops tuned from logged flight data

Projects

PowerSGD Gradient Compression for NanoGPT | *2-person course project, my extension*

- Built a data-parallel training setup with **PowerSGD** gradient compression for a 10.8M-parameter **Transformer** (**NanoGPT**) on NERSC **Perlmutter**, comparing ranks 1, 2, and 4 against **PyTorch DDP** on 4 and 8 **A100** GPUs over **NCCL**
- Cut the data each GPU sends per step by 59–156× but found steps ran 4–13% slower on one node, since the fast network made compression cost more than it saved; rank 4 matched DDP's best validation loss (1.475 vs 1.481) but needed about 3.3× as many steps

DeskPartner | *5-person team, Embodied Metal Hackathon, Honorable Mention* — github.com/aarochu/DeskPartner

- Architected the 5-module stack (arm control, vision calibration, **vision-language-model** planner, data collection, training) for a desk-cleaning robot arm and wrote the shared interfaces the team built against
- Built a perception-to-action agent loop where a vision-language model (**Claude Sonnet 4**) reads an overhead camera frame and emits a pick-and-place plan, with **ArUco** camera-to-arm calibration and an **OpenCV** fallback (validated in dry-run mode only)
- Wrote data-collection tooling for the team's vision-language-action policy (**MolmoAct2**): **LeRobot** episode recording, a dataset validator for missing cameras and extra channels, and a Rerun-to-LeRobot exporter

CUDATracker | *solo project* — github.com/aarochu/CUDATracker

- Built a real-time object detection and **SORT** tracking pipeline in **C++/Python** with custom **CUDA** preprocessing kernels validated against CPU output
- Benchmarked 9 configurations of **PyTorch**, **ONNX**, and **TensorRT** inference with per-stage latency tracing; FP16 TensorRT with CUDA preprocessing reached 77.1 FPS on a laptop GPU, while CUDA preprocessing did not help FP32 PyTorch because inference dominated

Technical Skills

Languages: Python, C++, C, CUDA
Deep Learning: PyTorch, PyTorch DDP, NCCL, Transformers (NanoGPT), TensorRT, ONNX, LeRobot, vision-language models
ML and Data: scikit-learn, XGBoost, NumPy, Pandas, OpenCV, Streamlit, AWS (S3, SQS)
Agentic AI Tools: Claude Code, Codex (skills, subagents, MCP servers, hooks), Anthropic API