

Shawn Liu

U.S. Permanent Resident, no sponsorship required

Research Engineer · neurosymbolic AI, secure ML inference, and LLM systems that ship
(510) 565-2523 · xl3704@columbia.edu · LinkedIn · github.com/ShawnnnLiu · OpenReview · shawnnnliu.github.io

EDUCATION

Columbia University · M.S. in Computer Science

Incoming Fall 2026 · Expected Jan 2028

University of California, Irvine · B.S. in Computer Science, GPA 3.844 / 4.0

Jun 2026

Dean's Honor List, all quarters (2022–2026).

PUBLICATIONS AND MANUSCRIPTS

Cross-Modal Event Encoder: Bridging Image–Text Knowledge to Event Streams

WACV 2026 (accepted) · arXiv:2412.03093

Liu, S. (co-author). Contributed the CLIP–event contrastive integration and the attention visualization pipeline (gradient-based attention maps and text relevance across N-ImageNet, N-Caltech101, UCF-Crime).

Integrating Symbolic & Neural Mechanisms for Adversarially Robust HDC

ISLPED 2026 (accepted) · DOI

Liu, S. (co-author). Contributed the robustness evaluation and the symbolic–neural encoding experiments.

Geometric Priors for Generalizable World Models via Vector Symbolic Architecture

NeurIPS 2025, NeurReps Workshop (accepted) · OpenReview

Liu, S. (co-author). Contributed the VSA representation experiments and ablations on the world-model grid; post-submission MiniGrid experiments.

Optimal Hyperdimensional Representation for Learning & Cognitive Computation

Frontiers in AI, 2026 (published) · paper

Liu, S. (third author). Contributed the encoding-dimension analysis, decoding-accuracy heatmaps, and signal-to-noise distributions.

RESEARCH EXPERIENCE

BiasLab, UC Irvine · Undergraduate ML Researcher (Advisor: Prof. Mohsen Imani)

Jun 2024 – Jun 2026

- Built a pose-estimation and symbolic-reasoning model in PyTorch for autonomous UAV carrier landing with the U.S. Navy. In adverse-weather simulation it cut landing failures, including deck-crew collision events, by 65% against the legacy fixed-pattern optical markers. The reasoning layer holds the landing whenever it sees a person on the deck. Dataset is CUI, so no visuals; the design can be walked through in detail.
- Aligned a trainable event-camera encoder to CLIP's frozen image–text space for zero-shot video anomaly detection: 57.7 / 64.1 / 54.4 AUC on XD-Violence / UCF-Crime / ShanghaiTech, +5.5 and +5.2 points over frame-based zero-shot CLIP on the latter two. WACV 2026.
- Characterized real CKKS-FHE noise in Microsoft SEAL under GNN operations and built a calibrated noise-injection simulation of encrypted graph inference; a distributed, noise-adaptive bootstrapping schedule cut projected bootstrapping overhead about 4x while holding over 90% accuracy in simulation.
- Implemented hyperdimensional computing encoders in PyTorch with the surrounding training and validation pipelines, plus Generalized Holographic Reduced Representations for encrypted-domain reasoning.

BioIntelligence Lab, UC Irvine · Undergraduate ML Researcher (Advisor: Dr. Haleh Alimohamadi)

Dec 2025 – Jul 2026

- Built a graph convolutional network over peptide 3D structure, one amino-acid residue per node, for graph-level property prediction, and a structure-aware minimum-inhibitory-concentration workflow fusing ESMFold-predicted structures with SVM geometric descriptors.
- Ran ablations isolating 3D conformation's contribution, curvature-only against curvature plus structure fusion, to quantify how peptide geometry drives antimicrobial potency. Ongoing: the activity versus hemolysis trade-off.

Softcom Labs, Cal Poly Pomona · Student Researcher

Dec 2020 – Apr 2021

- Built a real-time audio recognition and alarm system, a lightweight CNN over Mel-spectrogram input at 95% classification accuracy, deployed near-sensor on a Raspberry Pi with a Firebase and Flutter app on the Google Play Store.

ENGINEERING PROJECTS

Loop · LLM scheduling agent, live at loop-study.com Python 3.11, FastAPI, Pydantic v2, SQLite (WAL), React / TypeScript, Anthropic API, Google

Calendar OAuth, Docker, Fly.io

May 2026 – present

- Turns a career goal and weekly availability into a study plan written to Google Calendar. Five LLM nodes propose; deterministic code runs a five-stage validator (schema, graph, coverage, user-fit, scheduling), a human approval gate that rechecks the payload hash, duplicate detection, post-write verification, and rollback by stored event ID. The LLM is never handed a calendar.
- Built a two-tier eval over committed LLM recordings that never calls a model in CI: tier one gates schema validity at a floor of 1.0, post-repair invalid rate at a ceiling of 0.05, and citation coverage at a floor of 0.6444; tier two is an LLM judge that scores prose and never gates. Eight versioned eval sets of 6 to 16 cases, 12 committed recordings, one fixture that fails on purpose so the gate is known to catch failures.
- Pinned every prompt at two layers, SHA-256 of the system constant and SHA-256 of the full rendered frame, 14 pin cases across 7 nodes; a prompt-byte change without a version bump and a re-pin in the same commit fails make check. 4,826 backend and 326 frontend tests; 126 of 126 GitHub Actions runs green.
- Grounded plans in BM25 retrieval over a 242-document curated corpus, graded on 75 labeled queries at recall@5 0.7644, MRR 0.7144, nDCG@5 0.6910 with floors pinned to the measured values. Built and measured a Voyage dense-embedding hybrid with RRF fusion, MRR 0.8500 to 0.9311 on the 6-track corpus, and kept BM25 as the shipped retriever; a reranker was left unbuilt as a negative result by decision.
- 101 typed reason codes, an 18-state supervisor with 49 allowed transitions that raises on any other edge, LLM repair loops capped at two attempts, and an import-linter contract that keeps the LLM SDK out of every package except llm_nodes.
- In progress: partial syllabus regeneration (MVP regenerates fully) and per-user spend enforcement; the \$8 monthly cap and \$1.70 baseline cost are a written axiom with no enforcement in code, and no production usage is logged.

Foothold · transfer-credit navigator, live at foothold-transfer.com Python 3.12, FastAPI, Pydantic v2, SQLite FTS5, React 19 / TypeScript,

Anthropic API, Docker, Fly.io · Stellar Pathfinders 2026 entry

Jul 31 – Aug 21, 2026, solo

- Built the ASSIST articulation pipeline: 34,975 requests at one per second, zero 429s, an offline cache with an append-only manifest, and per-agreement fault isolation that turns any parse failure into a typed exclusion code. Output: 31,236 agreements and 352,024 articulation rows for 115 colleges and 15 UC/CSU campuses, committed as a 34.5 MB gzip.
- Wrote a deterministic evaluator over AllOf / AnyOf / course-leaf expression trees with nine typed finding codes; seven require a citation, two forbid one. No clock, no PRNG, stable ordering; a fixture recomputed in CI asserts the TypeScript triage board equals the Python one. Hand-verified against rendered assist.org pages on 2026-08-02.
- Counted N-from requirement groups by courses, not sections: the section-counting reading overstated 23,016 groups and turned pick-one-of-N into complete-all-N. Left ASSIST template overrides unmodelled because ASSIST's own renderer never applies them, so the artifact agrees with what a student sees.
- One Claude node drafts appeal letters; every course code and agreement key in the letter is validated against the findings bundle, and two failed repairs fall back to a deterministic template. Two-layer prompt pins; a call log whose contract makes a raw-prompt field impossible. 870 backend and 89 frontend tests, 298 invalid-input contract fixtures, mypy strict, 12 JSON schemas drift-checked in CI.
- In progress: a transcript-parser node is built and tested on the backend but has no frontend caller; agreements cover academic year 2025–2026 only with no refresh cadence.

Crash Anticipation · dashcam collision advisories PyTorch, VideoMAE (Hugging Face), YOLO11n, ByteTrack, OpenCV ·

github.com/ShawnnnLiu/Crash-Anticipation

Nov 2025 – Jul 2026

- Distilled a 21.88M-parameter VideoMAE-S teacher into a 2.95M-parameter MobileNetV2 + GRU student (alpha 0.5, temperature 2.0) whose causal recurrence makes each new frame one CNN forward: 2.45 ms per streaming step, 408 fps on an RTX 5070 Ti at batch size 1. On 300 CCD crash and 301 DAD no-crash videos: video AP 0.9998, 3.05 s mean warning before impact, 3 false alarms.
- Threw out the first benchmark after it scored AP 1.0 on a model that output p = 1.0 on every input, because both manifests contained only positives. Rebuilt it with 1,130 real no-crash videos, CCD temporal negatives, and time-to-event labels weighted by $\exp(-t_{te} / 1.2 \text{ s})$ inside a 1.5 s horizon with a 1.5 to 2.5 s guard band that contributes no gradient.
- Symbolic layer: looming as the least-squares slope of log box size over a 6-frame history (chosen for detector-jitter tolerance), a projected-miss test in object widths, time-to-collision only after the collision-course test holds two consecutive frames, and a rule engine that binds each advisory to a ByteTrack ID. 21 unit tests on synthetic pinhole scenarios; no quantitative benchmark for advisory correctness yet.
- In progress: ego-motion compensation and a metric bird's-eye-view state, both roadmap items with no code; no TensorRT or INT8 export.

Arrhythmia Classifier · MIT-BIH leakage study PyTorch, wfdb, scikit-learn · github.com/ShawnnnLiu/Arrhythmia_Classifier · CS 184A, team of 2

Nov 2025 – Aug 2026

- Showed that beat-wise splits leak patient identity on MIT-BIH: the same 64,966-parameter CNN scores 98.36% test accuracy beat-wise and 83.88% patient-wise, and a 443,527-parameter LSTM autoencoder drops from 98.95% to 82.06%. 48 records, 109,445 beats, 0.8 s windows at 360 Hz, six classes.
- Wrote a patient-wise split search: 50,000 sampled assignments out of 2,498,640,144, hard constraints that every class appears in the test set and that records 201 and 202 (one subject) never straddle a split, scored by KL plus L1 distance to the overall label distribution and a balance penalty. Result: val 13,716 and test 13,720 beats, a 4-beat imbalance, 3.53% total distribution difference.
- A leakage guard raises before training starts if any same-subject group spans two splits; macro F1 is reported over present classes alongside all-class macro, since absent classes score zero and conflate test composition with model quality. 13 pytest tests cover the split constraints and the metric.
- Cut a 3,349,190-parameter residual CNN that scored 72.3% to 81.5% on every patient-wise split, below the small CNN on each. In progress: no run yet on the corrected split, one seed per configuration, and an AAMI 5-class alignment proposed but not implemented. Writeup planned; no draft exists.

PROFESSIONAL EXPERIENCE**AdamsFoods · Full-Stack Contractor** (adamsfoodswholesale.com)

Summers 2024 and 2025

- Built and shipped a live wholesale e-commerce platform in React and Node / Express with S3 signed-URL media and JWT role-guarded admin routes, serving real customers.
- The internal livestock-distribution system, CRUD, search and filter, low-stock alerts, CSV export, 3D warehouse map, tracks 300+ SKUs and syncs to the storefront. It replaced handwritten logs and saves staff 5 to 7 hours a week.

Commit the Change, UC Irvine · Full-Stack Developer

Nov 2023 – Jun 2024

- Delivered pro-bono donation-tracking and authentication software for the nonprofit Feeding Pets of the Homeless, coordinating with chapter leads across a cross-functional agile team in React, Firebase, SQL, and Git.

TEACHING AND LEADERSHIP**Certified Learning Assistant, ICS 33: Intermediate Python · (Prof. Alex Thornton) · UC Irvine**

Apr 2025 – Jun 2025

- Certified peer educator (UNI STU 176, LA Pedagogy) supporting students in lecture and lab; weekly planning with faculty and TAs.

SKILLS**Languages:** Python, TypeScript / JavaScript, SQL, C / C++**ML / AI:** PyTorch, scikit-learn, CLIP · ViT · VideoMAE, knowledge distillation, graph neural networks, hyperdimensional computing, contrastive learning, ESMFold**Secure ML:** CKKS fully homomorphic encryption (Microsoft SEAL), noise characterization, bootstrapping-aware operator design, encrypted-inference simulation**LLM systems:** Anthropic API, structured outputs, bounded repair loops, two-layer prompt hash-pinning, recordings-based eval harnesses with CI gates, BM25 / FTS5 retrieval and retrieval eval (recall@k, MRR, nDCG), per-call cost and latency logging**Systems and infra:** FastAPI, Pydantic v2, SQLite (WAL, FTS5), PostgreSQL, React, Node / Express, REST APIs, Docker, Fly.io, AWS S3, GitHub Actions CI/CD, Google OAuth / JWT, Linux, CUDA**Spoken:** English (fluent), Mandarin (fluent)