

Damien Wang

Open to Relocation | Available Dec 2026 | +1 4127210501 | yunfanwang.us@gmail.com

Github: github.com/Yunfan-Wang | LinkedIn: linkedin.com/in/yunfan-wang-897276255

EDUCATION

Carnegie Mellon University — M.S. Information Networking

2025–2026

Focus: AI Systems, Distributed Systems, High-Performance Computing | INI Fellowship Scholarship

UNSW Sydney — B.E. + B.S. Computer Science (Hons double)

2020–2024

First Class Honours, GPA 3.91/4.0 | 3 times Dean's List 2021–2023 (Highly Commended)

TECHNICAL SKILLS

Programming: C++, Python, CUDA, Go

Machine Learning: PyTorch, TensorFlow, Multi-Agent Systems

Systems & Performance: AI system design, GPU acceleration, parallel computing, distributed systems

SELECTED PROJECTS

Scalable Multi-Agent Market Simulator (pittsai.com/sams)

Pittsburgh.PA/Berkeley.CA, USA

Lead Architect & Developer | Python, PyTorch, Distributed Systems

7/2026-9/2026

- Designed a market-simulation architecture that separates a shared Transformer-based market interpreter from independent stateful decision agents, to realize more efficient, fluctuation enduring market simulator.
- Developed an original end-to-end **PyTorch training** architecture on **Optiver datasets**. Engineered **19** normalized microstructure features into **129-step sequences**, encoded them with a 4-layer, 8-head causal Transformer (d=128), and advanced 24 independently parameterized GRU agents with private recurrent state and stochastic actions; learned softmax population gate aggregates agent actions into a probabilistic transition head for prediction
- Trained the full **encoder-agent-gate-transition stack** jointly using **Gaussian NLL + agent-diversity** and **gate-balance regularization**, **AdamW/OneCycle scheduling**, mixed precision, gradient clipping, bounded log-variance and finite-gradient checks; while supporting recursive stochastic state reconstruction for multi-step simulation.
- Deployed a public **forward multi-agent market-simulation demo** at pittsai.com/sams, running **24 recurrent agents**.
- Early stable experiments demonstrated high-60% one-step directional structure and authored the technical paper; **placed 2nd in the AI Innovation track** and presented at the **BigQuant 2026 Global Competition North America Final** at UC Berkeley.

Distributed Backtesting & Market Data System

Pittsburgh.PA, USA

Project Owner, Algorithm Developer | Go, C++

3/2026 - 6/2026

- Architected a distributed historical-market replay and backtesting system with a **metadata coordinator/control plane**, **time-partitioned chunk servers**, **execution workers** and **client API/CLI**; used internal RPC to separate job/control metadata from replay execution and data-serving paths so strategies could run concurrently without centralizing the hot data plane.
- Designed chunk **lifecycle** and **recovery** around **lease-owned writable partitions to immutable sealed chunks**, **replicas**, **operation logs** and **checkpoints**; added metadata caching and deterministic ordered replay to guarantee reproducible event sequences.
- Separated a **Go control/scheduling layer** from **performance-sensitive C++ replay/strategy execution**, defining **worker registration**, **heartbeat/failure recovery** and **chunk-oriented** replay interfaces to model the coordination, storage and fault-tolerance tradeoffs of production quantitative research infrastructure. Rated 340/340 by project supervisor according to the performance list.

High-Performance Option Pricing Engine

Pittsburgh.PA, USA

Project Proposer & Developer | CUDA, C++

8/2025 - 12/2025

- Built a GPU-accelerated **Longstaff-Schwartz Monte Carlo** engine for American option pricing over **50k+ simulated paths**, parallelizing path generation, payoff evaluation, CV estimation and backward exercise decisions on CUDA-capable hardware.
- Reworked memory/execution around **coalesced path layouts**, **batched random-number generation**, **fused payoff kernels** and **parallel reductions**; minimized host-device synchronization and redundant intermediate transfers across exercise dates
- Achieved **60x speedup over the CPU baseline** while preserving numerical agreement within **~1% pricing error**; used bootstrapping and variance-reduction experiments to validate estimator stability and quantify Monte Carlo uncertainty.
- Received **100/100** for the CMU parallel-computing coursework project and used profiling-driven CUDA optimization to turn the serial numerical-finance algorithm into a scalable GPU workload.

Multi-Agent Philosophic ML Decision System (OUSI.AI)

Pittsburgh.PA, USA

Founder & ML Engineer | Python, PyTorch, Full Stack

8/2025 - Present

- Founded and built a full-stack stateful AI product with a **FastAPI + SQLAlchemy** backend and **Next.js/React** frontend; implemented **authentication/user synchronisation**, **entitlements**, **interaction logging** and **product APIs** supporting **8 distinct interaction modes** including **rapid/async response**, **multi-agent councils**, **guided reflection**, **character simulation** and **long-form deliberation**.
- Designed a modular **multi-agent orchestration** layer in which **12 philosopher/persona agents** generate independent perspectives, /debate one another and feed an aggregation/synthesis stage; parallelised independent LLM calls for bulk user calls.
- Built persistent conversational state across sessions using conversation/message history, user progress, mood/context signals, feedback, interaction replay and **50-message rolling memory**.
- Implemented the product as a reusable AI-systems platform, separating efficient model providers, persona logic, state storage and user-facing workflows. Accepted by **The Swartz Center for Entrepreneurship**, **100+ pilot users**, with good retention evidence.

SELECTED INTERNSHIP

VortexNet

Covina, CA, USA

AI Systems Engineering Intern / Project Lead

5/2026 - 7/2026

- Led development of a **multi-tenant enterprise AI knowledge platform** for a financial institution, integrating **FastAPI**, **PostgreSQL**, **S3** and **JWT/RBAC** to serve **~10K+ documents / 100+ users** with tenant-isolated access.
- Built the ingestion and retrieval stack for **PDF/Word/Markdown/TXT**, combining semantic chunking, batched embeddings, **pgvector HNSW + PostgreSQL keyword search + cross-encoder re-ranking**
- Designed a pluggable **LLM/RAG orchestration layer** supporting OpenAI and local **vLLM** models, citation verification, query rewriting and low-confidence fallback.
- Engineered **Celery + Redis async processing**, 3-layer caching, audit logs and **Dockerized AWS CI/CD**; cut **P95 query latency** by **~40%** and external LLM/embedding API calls by **~50%** under benchmark workloads.

UBER

Hong Kong SAR, China

Algorithm Developer

12/2022 - 2/2023

- Built an end-to-end **real-time gesture-recognition pipeline** covering collection, annotation, preprocessing, temporal feature extraction, training and inference across **~20K labeled sequences / 12 gesture classes**.
- Implemented **keypoint-based spatiotemporal modeling** with **CNN + LSTM/TCN**, temporal smoothing, coordinate normalization and augmentation; improved **macro-F1** from **~0.78** to **~0.89**.
- Ran controlled experiments over sequence length, feature normalization, augmentation and model depth; reduced **false gesture triggers** by **~35%** and improved robustness across users, lighting and viewpoints.
- Optimized inference through landmark-first processing and batched model execution, reaching **~15–20 ms/sequence** on target hardware; modularized training/evaluation scripts for rapid addition of new gesture classes.