

Yuliang Peng

ypeng12@umd.edu | [ypeng12.github.io/MyPage/](https://github.com/ypeng12) | [linkedin.com/in/ypeng12](https://www.linkedin.com/in/ypeng12) | 240-886-9716

EDUCATION

University of Maryland, College Park

M.S. in Applied Machine Learning

GPA: 3.8/4.0, Expected May 2027

B.S. in Computer Science

GPA: 3.6/4.0, May 2024

Relevant Coursework: Data Structures and Algorithms, Operating Systems, Systems for Machine Learning, Cloud Computing

WORK EXPERIENCE

ICBC USA — *Software Engineer Intern* | *Financial Operations Systems* | Los Angeles, CA | Jun 2026 - Aug 2026

- Built **Python/SQL** services for exception-processing workflows across four digital-banking operations, validating transaction and document inputs and routing cases through retry, review, escalation, and resolution states.
- Designed SQL-backed case-state and audit models for **200+** financial exceptions, capturing lifecycle transitions, failure reasons, reviewer actions, and compliance outcomes for end-to-end traceability.
- Developed reconciliation and monitoring workflows for deposit and card-payment failures, reducing repeated case lookup and follow-up effort by **20%** while improving operational visibility and auditability.

ModeSens — *Software Engineer (Full-time)* | *Data Infrastructure* | Remote, USA | Apr 2025 - Sep 2025

- Built **Python/AWS** data-ingestion pipelines for **30+** external retailer feeds, normalizing heterogeneous product, pricing, availability, image, and purchase-link schemas into unified catalog records.
- Productionized ingestion and matching workflows with SQL-backed audit, reconciliation, validation, and **CI/CD** checks, reducing processing failures by **20%** and manual duplicate review by **80%**.
- Developed a multi-stage matching pipeline combining depth-derived shape embeddings, perceptual hashing, and SSIM, improving match accuracy by **25%** across noisy external data sources.

Bank of China — *Software Engineer (Full-time)* | *Transaction Processing & Reconciliation* | Los Angeles, CA | Jul 2024 - Apr 2025

- Built **Java/Spring Boot** transaction-processing and reconciliation services exposing APIs for transaction status, retry history, failure reason, settlement state, and downstream reporting.
- Implemented PostgreSQL transaction models with idempotency keys, lifecycle checks, and audit trails to detect duplicate requests, invalid state transitions, and missing settlements.
- Resolved **70+** production incidents through log analysis, root-cause isolation, and pipeline fixes; added Grafana monitoring for p95 latency, reconciliation gaps, retries, and failed jobs, reducing incident-triage time by **30%**.

PROJECTS

[InferRoute](#) | *LLM Inference Gateway & Control Plane* | **Python, FastAPI, Redis, PostgreSQL, Docker**

- Built a multi-provider inference gateway integrating OpenAI, Gemini, vLLM, and Ollama, with request validation, streaming responses, timeout enforcement, credential isolation, and containerized deployment.
- Implemented TCP Vegas-inspired adaptive concurrency control and sliding-window circuit breakers; sustained **45.2 RPS** at 100 concurrent clients with **99.4%** success and **120 ms** p95 gateway overhead under provider failures.
- Built load-testing and failure-injection workflows for tail latency, failover, and recovery; evaluated 3.2M WildChat conversations, reducing estimated API cost by 54.2% while retaining 98.8% of baseline quality.

[Quant.ai](#) | *Low-Latency Market Replay & Execution System* | **C++17, Linux, POSIX Sockets, TCP/UDP, pybind11, Python**

- Built a deterministic event-driven market replay and backtesting system with a C++17 price-time-priority limit order book supporting amendments, cancellations, partial fills, risk checks, and sequence validation.
- Built **UDP market-data ingestion** and **TCP order-entry** components over POSIX sockets with binary message framing, packet sequencing, gap detection, acknowledgments, timeouts, and reconnect recovery.
- Profiled hot paths with perf and gdb and validated memory safety with AddressSanitizer and UndefinedBehaviorSanitizer; processed **2.50M** events at **507,405** events/sec and **3.8 μs** p99 latency, with all determinism and invariant checks passing.

PUBLICATION

International Cargo Charter Operations and Preighter Performance | *Transport Policy*, Vol. 184, Article 104173, 2026

- Co-authored a study on CAAC cargo-charter networks, contributing software, data curation, validation, and visualization.

SKILLS

Languages: Python, C++, Java, JavaScript/TypeScript, SQL | **Spoken:** English, Chinese, Japanese

Backend / Infrastructure: Spring Boot, FastAPI, PostgreSQL, Redis, Docker, AWS, Grafana, CI/CD, Git

ML / Data: PyTorch, TensorFlow, Hugging Face, Pandas, NumPy, Selenium