Immanuel Peter

ipeter@uchicago.edu | (479) 257-3842 | linkedin.com/in/immanuel-peter | github.com/immanuel-peter | ipeter.dev

EDUCATION

The University of Chicago, Chicago, IL
Bachelor of Science in Computer Science, expected June 2028
Relevant Coursework: Intro to Computer Science, Systems Programming, Mathematical Foundations of Machine Learning, Abstract Linear Algebra, Analysis in Rn I, Discrete Mathematics, Introduction to Complexity Theory

SKILLS

Software Engineering: Python, TypeScript, React, Node.js, Docker, PostgreSQL, Go, REST APIs, Cursor, Codex
AI/ML Engineering: PyTorch, JAX, Transformers, RAG, NLP, Computer Vision, Model Training, Hugging Face, LLMs, OpenAI API

EXPERIENCE

Quantum Rings, Chicago, IL, *Software Engineer Intern*, June 2025 – August 2025

- Built full-stack admin analytics dashboards (NestJS, Next.js, MUI/Recharts) with time-bucketed metrics and KPI visualizations, enabling leadership to monitor user growth, execution trends, and marketing attribution in real time.
- Engineered queue-driven background workers (AWS SQS, TypeORM) for telemetry aggregation, execution processing, and HubSpot sync, unlocking scalable, fault-tolerant data pipelines and reducing API latency.
- Designed developer-facing features including public profile pages, LinkedIn certification provisioning, and execution widgets, strengthening community engagement and platform credibility.
- Shipped 19 PRs across 15 assigned issues (≈15K LOC added, 3.6K deleted), contributing 43 commits with cross-stack code reviews and schema refactors that improved backend stability and modularity.

PROJECTS

Matchbox: AI-Powered Research Lab Matchmaking Platform

- Engineered a full-stack web platform for AI-driven student–lab matchmaking using Next.js and FastAPI, implementing dynamic user interfaces and RESTful APIs for profile management, lab postings, and student applications.
- Implemented a retrieval-augmented matching pipeline integrating Chroma (vector database) for semantic search and OpenAI GPT for LLM-based scoring, delivering personalized lab recommendations with AI-generated fit scores.
- Architected cloud infrastructure and deployment on Google Cloud using Docker containers and Terraform, provisioning services (Cloud Run, Firestore, Cloud Storage) for a scalable, secure production environment.
- Implemented CI/CD pipelines (GitHub Actions) for automated testing and deployment, streamlining build processes and ensuring rapid and reliable release cycles.
  Tech Stack: Python, FastAPI, Next.js, TypeScript, Tailwind CSS, Docker, Terraform, Google Cloud (Cloud Run, Firestore, Cloud Storage), Chroma, OpenAI API

AutoMoE: MoE Self-Driving Model

- Architected and implemented a Mixture-of-Experts framework in PyTorch for autonomous driving, integrating specialized deep learning models for object detection, semantic segmentation, and end-to-end trajectory prediction.
- Trained and validated the system on large-scale, diverse datasets, including BDD100k and nuScenes, as well as in the CARLA simulator, to ensure robust performance across various driving scenarios and sensor modalities (camera and LiDAR).
- Developed a dynamic gating network to efficiently select the optimal "expert" model in real-time, enabling multi-task learning and efficient inference for complex driving environments.
  Tech Stack: Python, PyTorch, PyTorch DDP, CARLA, Docker, Bash, Linux, NumPy, Hugging Face, Nvidia Brev

CARLA Autopilot Datasets (Open Source)

- **CARLA Autopilot Images**: multi-camera dataset (68K frames, ~188 GB) with synchronized ego state + controls for vision-to-control and imitation learning.
- **CARLA Autopilot Multimodal**: extended version (82K frames, ~365 GB) adding semantic segmentation, LiDAR, 2D bounding boxes, and richer environment metadata for sensor fusion and RL research.
  Tech Stack: Python, CARLA, Hugging Face Datasets, NumPy, Linux

LocalRAG: Terminal LLM with Infinite Memory

- Developed a command-line interface (CLI) for seamless, ChatGPT-style interactions with large language models
- Integrated a local FAISS vector database for conversational memory, enabling contextual responses across sessions
  Tech Stack: Python, FAISS, Sentence Transformers, OpenAI API, Anthropic API, Click, Rich