Vincent Zhong

vincent.zhong@uwaterloo.ca | github.com/vincentzed | linkedin.com/vincentzed | 613-701-2828

Education

University of Waterloo — Sep 2024 – Apr 2029 (Expected)
Electrical and Computer Engineering, BASc — *Waterloo, ON*

- Computer Science Club Projects Mentor

Experience

Machine Learning Engineer Intern — Sep 2025 – Dec 2025
Shopify — *Toronto, ON*

- Building query rewriting and inference systems for production search infrastructure serving 2M+ monthly queries; optimizing retrieval and query rewriting performance and quality
- WIP

Software Engineering Intern — Feb 2025 – May 2025
Yupp AI — *Mountain View, CA*

- Built and deployed embedding service using Nomic Embed for user memory retrieval, handling 200K queries/day on GCP (FastAPI, PyTorch, Docker)
- Sped up CI/CD pipelines for ML microservices by cutting build time from 22 to 12 min and reducing time-to-merge by 30% (Docker, GitHub Actions, GCP)
- Deployed vector retrieval system indexing 1M user memory embeddings with pgvector

Systems Researcher — Nov 2024 – May 2025
Software Systems at UW Data Systems Group — *Waterloo, ON*

- Co-author: ”Retrieval with Learned Dense and Sparse Representations Using Anserini” (ACM SIGIR 2025 Resource & Reproducibility Track)
- Integrated Snowflake’s Arctic text embedding models into a retrieval research toolkit as a new dense encoder using ONNX, implementing tokenization and inference to enable embeddings for passage retrieval benchmarks.
- Added support for a new, efficient sparse encoding model using ONNX to Anserini, boosting query relevance performance by up to 10% on the leading in-domain benchmark.
- Developed a search and conversational interface using Vercel’s AI SDK, allowing users to query dense and sparse indexes using natural language, while also displaying query relevance metrics.

Projects

Parallel Deep Learning Techniques | *PyTorch*

- Implemented data parallelism from scratch, featuring model replication to GPU instances, data sharding, and gradient synchronization (CPU averaging).
- Designed parallelized forward/backward passes and inter-stage activation/gradient propagation.
- Validated correctness and convergence against PyTorch’s `nn.DataParallel` and sequential execution baselines.

Technical Skills

Languages: Python, Java, C++, TypeScript, SQL, Bash
Frameworks: PyTorch, FastAPI, React, Next.js
Tools: Git, Docker, GitHub Actions, Redis, PostgreSQL, Google Cloud Platform