Nguyen Le

lenguyen18072003@gmail.com +84-942-142-797 Website Github

Selected Articles

- **Finding and Fighting the Lazy Unlearner: An Adversarial Approach** (Details): Investigated a failure mode of RMU unlearning where models collapse to a single "lazy" direction that encodes evasion (extension of my thesis). I show how to find that direction, analyze its semantic composition with Sparse Autoencoders (SAEs), quantify its causal effect, and propose an adversarial regularizer to force more robust unlearning.
- **Residual Stream is Key to Transformer Interpretability** (Details): A technical note and explanation for A Mathematical Framework for Transformer Circuits paper.
- **PRML for Viet** (Website): An attempt to provide a resource/reference for Vietnamse student to learn about Machine Learning in general and learn Pattern Recognition and Machine Learning book in specific.

Education

- **VNUHCM - University of Science, Viet Nam** | Oct 2021 - Oct 2025 | Bachelor of Science | Major in Artificial Intelligence | **GPA**: 8.5/10 (3.4/4).

Experience

- **VinBigdata - Ha Noi, Viet Nam** | July 2025 - Current | AI Engineer.
- **Gameloft Vietnam - Ho Chi Minh, Viet Nam** | November 2024 - March 2025 | C++ Game Programmer Intern | Implemented and shipped new 2 UI features for Asphalt 8 (VIP UI, Gacha system UI), contributing to a game with over 100M downloads.
- **Bosch Global Software Technologies Vietnam - Ho Chi Minh, Viet Nam** (Details) | August 2024 - November 2024 | AI Engineer Intern | Developed a full-stack RAG application, featuring a VSCode extension front-end and a FastAPI/Langchain back-end, to automate fuzz test generation for C source code.

Selected Projects

- **gemm_metal** (Github): Achieved 421 GFLOPS (≈ 17% peak GFLOPS of M2 chip) by optimizing shared memory access with 2D tiling block and SIMD Group. This represents a ≈3x performance increase over a naive baseline (171 GFLOPS). **Tech stack**: C++17, Metal (Apple's GPU).
- **banhxeo** (Github): A simple NLP library built with Python and Jax/Flax. Implement NLP models (RNN, MLP, etc.) and Tokenizer system from scratch.
- **yamc** (Details): Developed a simple neural network from scratch with simple custom matrix operations in C++ to classify MNIST, CIFAR-10, etc. datasets. Architecture supports configurable layers (Convolution, Dropout, Linear, etc.).

Skills

Programming Languages: C++, Python | **System Programming**: OpenGL, Metal, CUDA, Triton | **Frameworks**: Pytorch, Jax | **Tools**: Git, Unix Shell, Docker, Latex, CMake.

Languages

English: Professional working proficiency | *TOEIC Reading & Listening*: 870/990, *TOEIC Speaking & Writing*: 320/400, Details.