Nguyen Le

lenguyen18072003@gmail.com +84-942-142-797 lenguyen.vercel.app LinkedIn GitHub

Work Experience

- **VinMotion – Ha Noi, Viet Nam** *December 2025 – June 2026*
  Robotics Engineer (Internship)
  - Integrated and optimized RL policies in a real-time humanoid control stack, debugging latency and hardware-interface failures during deployment on physical robots.
- **VinBigdata – Ha Noi, Viet Nam** *July 2025 – December 2025*
  AI Engineer (Fellowship)
  - Selected as one of 70 participants from 900 applicants for a fully funded AI engineering fellowship. Completed an intensive training program in CV, NLP, and generative AI; delivered projects supervised by senior researchers from Vietnamese universities and industry.
- **Gameloft Vietnam - Ho Chi Minh, Viet Nam** *November 2024 – March 2025*
  C++ Game Programmer (Internship)
  - Engineered and shipped a new UI subsystem for Asphalt 8.
- **Bosch Global Software Technologies Vietnam - Ho Chi Minh, Viet Nam** *August 2024 – November 2024*
  AI Engineer (Internship)
  - 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

Education

- **VNUHCM – University of Science, Ho Chi Minh, Viet Nam** *Oct 2021 – Oct 2025*
  B.Sc. Artificial Intelligence | Graduated with *Very Good Degree* **GPA**: 3.4/4.0 (8.51/10)

Project

- **comtam** – *A tiny Deep Learning framework for Apple Silicon* Github
  - Built an eager-mode tensor runtime for Apple GPUs with explicit device, storage, tensor-view, command-dispatch, and Metal pipeline-cache abstractions. Implemented move-safe MTL::Buffer ownership, shared tensor storage, typed host-device transfers, and runtime dispatching of custom float32 Metal.
- **cuda_lab** – *A tiny lab for experimenting with CUDA/Triton kernels* Github
  - Aggressively optimized GEMM to reach up to 80% of cuBLAS FP32 GEMM throughput on an RTX A6000 for selected matrix shapes by applying wraptiling and tensor core utilization; also implemented LLM-related kernels: LayerNorm, Softmax, and Flash-Attention 2.
- **banhxeo** – *A tiny custom Tensor Compiler* Github
  - Built a lazy tensor runtime with stride-based views and automatic differentiation; implemented graph scheduling and Triton code generation that lowers fused unary, binary, ternary, and view operations into single JIT-compiled GPU kernels.

Research Experience

- **Fantastic Directions and Where to Find Them: Dissecting the Lazy Mechanism Inside RMU**
  Undergraduate Thesis Link
  - Investigated "shallow unlearning" in RMU on Gemma-2-2B; demonstrated via mechanistic interpretability that LLMs mask rather than erase harmful knowledge.

Technical Writings

- **Why Do We Need KV Caching?** Link
- **Residual Stream is Key to Transformer Interpretability**
  - In-depth research note on mechanistic interpretability of Transformers, covering residual stream theory, virtual weights, and attention head behavior.

Skill

- **Programming Languages**: C++, Python.
- **Framework**: PyTorch, JAX, HuggingFace-ecosystem (Transformers, TRL, etc.), IsaacSim/Lab, Mujoco.
- **Systems**: CUDA, Metal, ROS, Triton.
- **Tools**: Git, Linux, Docker, CMake, LaTeX.

Languages

English: Professional working proficiency