Naveenraj Kamalakannan

New York City, NY · +1 914-490-3063 · naveenraj.k@nyu.edu
itsnav.com · github.com/therealnaveenkamal · linkedin.com/in/navz/

Education

New York University (NYU) — Brooklyn, NY
M.S. in Computer Engineering | ***GPA: 3.96 / 4.00*** — *Sep 2024 – May 2026*

- Coursework: High-Performance ML, Deep Learning, Computer Vision, Reinforcement Learning, Distributed Systems.
- Research interests: LLM inference & training systems, multimodal reasoning, human motion understanding, evaluation & interpretability.

Vellore Institute of Technology (VIT) — Vellore, India
B.Tech. in Electronics and Communication Engineering | *GPA: 8.79 / 10.0* — *Jul 2018 – May 2022*

Open Source & Independent Projects (LLM Systems)

NVFP4 W4A4 Quantization & Serving — Nemotron-3.5-Lightning-30B-A3B — Python, CUDA, vLLM
Independent Project | *HF: nemotron35-w4a4-v11* — *2026*

- Produced a near-lossless **NVFP4 W4A4** quantization of NVIDIA's Nemotron-3.5-Lightning-30B-A3B (hybrid MoE + Mamba) with TensorRT Model Optimizer — FP4 activations on routed-expert GEMMs, FP8 shared experts and Mamba projections — running expert GEMMs on Blackwell's native FP4 tensor cores to cut **TTFT by ∼32–34%** vs. NVIDIA's official W4A16 release at only +1.2% perplexity.
- Built **mcond**, a vLLM plugin implementing **M-conditional kernel dispatch** (Marlin at decode, CUTLASS FP4 at prefill) via dual `FusedMoEKernel` instances resolved at CUDA-graph capture for zero steady-state dispatch cost; final config beats the official W4A16 checkpoint on **every measured metric**: 4.5% lower TPOT, 30% lower TTFT, +8% single-stream throughput.

vLLM — Python, CUDA, PyTorch
Contributor | *PR #34823* — *Remote*

- Refactored **Multi-Head Latent Attention (MLA)** to decouple prefill/decode paths from a unified custom op, enabling `torch.compile` fusion and piecewise CUDA Graph capture, reducing Python overhead and making MLA more modular and easier to experiment with.

Microsoft DeepSpeed — Python, Distributed Systems
Contributor | *PR #8204, PR #7302* — *Remote*

- Implementing a **DeepCompile compiler pass for AutoTP** (PR #8204, in review): introduced `copy_to_tp` / `reduce_from_tp` collective primitives and FX-graph rewrites that identify column/row-parallel matmuls via `nn_module_stack`, deferring tensor-parallel communication into the compiled graph.
- Fixed a critical **ZeRO-3 CPU-offload gradient clipping bug** (PR #7302), ensuring global gradient norms correctly reflect clipped gradients during offload scenarios and improving training stability for large models.
- Collaborating with Olatunji Ruwase on PyTorch Core (Issue #158187) to implement Zip serialization support for **DeepNVMe**, improving I/O throughput and compatibility for NVMe-offloaded checkpoints.

Snowflake ArcticInference — Python, vLLM Plugin
Contributor | *PR #124* — *Remote*

- Integrated **FlashInfer** backend support into SwiftKV, optimizing high-throughput KV-cache-aware decoding and enabling more efficient large-scale LLM serving within the ArcticInference stack.

Professional Experience

J.P. Morgan (Asset & Wealth Management) — New York, NY
AI/ML Associate — *July 2026 – Present*

- Building agentic AI applications for investment workflows, taking multi-agent systems from prototype to scalable production deployments on the firm's agentic platform.
- Driving performance engineering of the LLM serving stack: latency/throughput profiling, inference optimization, and capacity planning for production agentic workloads.

J.P. Morgan (Asset & Wealth Management) New York, NY
AI & Data Science Associate Intern *Jun 2025 – Aug 2025*

- Architected an agentic platform using multi-stage retrieval pipelines (RAG), context engineering, and rerankers to reduce latency and improve relevance in financial LLM tools.
- Orchestrated multiple AI agents with advanced reasoning capabilities and a shared memory layer, integrated via **MCP servers** into production workflows for complex investment use cases.

Zeeco Middle East Dammam, Saudi Arabia
Control Engineer *Jul 2022 – Aug 2023*

- Engineered an anomaly detection system for circuit designs using neural networks, reducing manual verification effort by ∼**40%**, and designed industrial control strategies (PLC/DCS) with predictive models for Pressure, Temperature, and Flow optimization, improving efficiency by ∼**30%**.

Research Experience

NYU Center for Data Science & NYU Langone New York, NY
Research Assistant (Advisors: Prof. C. Fernandez-Granda, Prof. H. Schambra) *Feb 2025 – May 2026*

- **Video Action Pipeline for Stroke Rehabilitation.** Studied whether current VLMs can recover clinically meaningful upper-limb motion primitives from rehab videos. Benchmarked SOTA VLMs (InternVL3, NVILA, LLaVa-OneVision) and engineered a pose-refined prompting pipeline that integrates YOLOv11 pose tracks with VLM context to extract sub-second motion primitives, achieving ∼67.75 Edit Score (ES) on structured upper-limb rehab tasks and exposing systematic failure modes on subtle hand-object interactions.

Selected Publications

Li, V., **Kamalakannan, N.**, et al. "The Potential and Limitations of Vision-Language Models for Human Motion Understanding." *arXiv preprint arXiv:2511.17727*, 2025. [Link]

Kamalakannan, N., et al. "Exponential Pixelating Integral Transform with Dual Fractal Features for Enhanced Chest X-Ray Abnormality Detection." *Computers in Biology and Medicine*, Vol. 182, 2024. [Link]

Technical Skills

Languages: Python, C++, CUDA, Bash, SQL
ML Systems: vLLM, DeepSpeed, TensorRT-LLM, TensorRT Model Optimizer, PyTorch Distributed (DDP/FSDP), `torch.compile` / CUDA Graphs, FlashInfer, Marlin & CUTLASS kernel routing, KV-cache optimization
Quantization: NVFP4, FP8, W4A4/W4A16 recipes, PTQ calibration (ModelOpt)
Model Architectures: Transformers, Vision–Language Models (VLMs), Mixture of Experts (MoE), Mamba hybrids, SAM-2
Concepts: Deep Learning, Optimization & Evaluation of LLM/VLM Systems, Mechanistic Interpretability, Reinforcement Learning, High-Performance Computing
Tools: Docker, MLflow, Git, Weights & Biases, Nsight Systems, Linux