Narain Pattabhiraman

AI/ML Infrastructure Engineer | LLM Serving, Agentic Systems, GPU Inference, MLOps

Linkedin Github Portfolio narain13579@gmail.com

EDUCATION

Arizona State University, *M.S, Computer Engineering, Computer Systems* *Aug '23 – May '25*

Amrita Vishwa Vidyapeetham, *B.Tech., Mechanical Engineering* *Jul '16 - Jun '20*

SKILLS AND CERTIFICATIONS

Programming & Systems: Python, C/C++, CUDA, Triton, SQL, Kubernetes, Terraform, Docker.
LLM Systems: vLLM, SGLang, GGML, DeepSpeed, LoRA, SFT, GRPO/RLHF, RAG.
Agentic AI & Evals: ReAct, MCP, Composio, DAG Workflows, Tool Use, Langfuse, lm-eval-harness, Agentic Evaluation.
ML & Cloud: PyTorch, Computer Vision, NLP, Model Optimization, A/B Testing, AWS, GCP, Azure, OpenSearch, MLflow.

WORK EXPERIENCE

MLOps Engineer, ASU Enterprise Technology, Tempe, USA *Jan '24 – Present*

- Built **CreateAI**, an agentic LLM platform serving **30K MAU** and **30B+ tokens/month** across **100+ provider/in-house models** with multi-model routing, custom **RAG**, semantic caching, and hallucination defense.
- Architected **Agentic Workflows**, upgrading query Lambda/DAG workflows to a **ReAct agent runtime** powering ∼**100K jobs/month** with tool execution, **MCP**/Composio, multi-agent handoffs, sandboxed code execution, asset handling, and agentic history.
- Built LLM serving infrastructure on ASU's **SOL Cluster with Kubernetes**, enabling dynamic GPU resource scaling for multi-tenant inference workloads.
- Deployed **serverless LLM inference on AWS Lambda using GGML**, reducing inference cost by **60% vs AWS Bedrock** while enabling custom-model deployment at scale.
- Trained domain-specific **embedding models** via MLM pretraining on **DeBERTa/OpenSearch**; extended RAG to a **serverless multimodal indexing system** for text/video/image QA, processing typical files in ∼15s avg with robust DOCX/PDF extraction edge cases.

Software Engineer – ML, Fidelity Investments, Chennai, India *Apr '22 – Jul '23*

- Deployed **BLOOM** and **Flan-T5** in production on **4× A100 GPUs** with **tensor/model parallelism**, sustaining **40 tokens/sec at 8 concurrent requests** for low-latency LLM inference.
- Built a **low-touch model deployment pipeline** on **AWS**, cutting adoption time from **3 months to 2 weeks** – shipped **15 models to production in the first quarter** post-release.
- Designed end-to-end **MLOps and CI/CD self-service** pipelines for LLM inference, reducing deployment turnaround by **40%**.
- Engineered **online feature stores** for real-time inference and optimized **batch inferencing** at scale, including **Document Layout Models**.

AI Researcher, QPIAI Technologies, Bangalore, India *May '21 – Apr '22*

- Built a scalable **CV/NLP inference platform** across multiple machines with queue-based scheduling, dynamic batching, online inference, and batch jobs for edge and server workloads.
- Led **hardware/software co-design** and multi-GPU training/inference optimizations to maximize throughput on target accelerators.
- Built a **Graph Convolutional Network** for circuit-layout similarity matching and an **AI-driven extraction tool** that parsed engineering documents into graph representations for downstream retrieval.

Data Analyst, Thoughtware Analytics, Bangalore, India *Jan '20 – May '21*

- Built **Computer Vision** systems for manufacturing QA and **Demand Forecasting** models for supply-chain clients, driving waste reduction and logistics optimization.

Open Source Contributions

- Contributed to **PyTorch** autograd kernels, **Kornia** GPU image decoding, **SGLang** inference configuration docs, and **vLLM** LoRA support for Apple **MLX** runtime.

PUBLICATIONS

- **Large Language Model Inference Optimizations** *DISML 2024*

PROJECTS

Reasoning-Tuned LLM via Synthetic Data and GRPO *Winner, AMD Synthetic Data Hackathon*

- Built a multi-stage **synthetic data generation pipeline** producing JSON-formatted reasoning QA pairs for logic-puzzle tasks via structured relationship sampling, entity assignment, and bidirectional validation.
- Fine-tuned **Qwen3 14B** with **LoRA + SFT**, then post-trained with **GRPO** using weighted multi-component rewards (correctness, JSON format compliance, completeness) to improve reasoning-chain quality.
- Engineered the end-to-end **RL training environment** – sampling synthetic problems, running rollouts, and applying programmatic verifiers as reward signal for sample-efficient training.

Compiler System for Graph Optimization and Op-Fusion for Deep Learning

- Engineered a graph-optimization compiler for mobile backends using **TVM** and **TASO**; wrote **ARM NEON** SIMD and custom **CUDA** kernels (**Jetson Orin**) with measurable speedups over OpenCV C++.