Akhil Shekkari

+1 4254268292 | shekkari.akhil@gmail.com | linkedin | github | portfolio

Summary

AI Engineer with 3+ years of production experience specializing in Inference optimization , RAG pipelines, multi-agent orchestration, and LLM evaluation, agentic systems built LangGraph-based agents with tool use, multi-step planning, LangSmith observability; MS in Applied ML from UMD.

Projects

DeepSeek-R1-Style Reasoning Pipeline GitHub **2026**

- Post-trained Qwen2.5-3B with a pipeline built from scratch, reproduced the DeepSeek-R1 workflow in raw PyTorch: SFT on a 10k-subset of OpenR1-Math chain-of-thought traces followed by **Dr.GRPO** reinforcement learning with group-relative advantages, clipped policy loss, and verifiable exact-match rewards; no critic model required.
- Scaled with **FSDP** across 2× 96GB Blackwell GPUs; conducted memory sweeps across batch size and sequence length, reaching 86% GPU utilization before RL training.
- Evaluated end-to-end on **GSM8K** and **MATH500** using a consistent CoT prompt harness; SFT lifted base accuracy from 63.2% to 71.2% on GSM8K and 26.3% to 33.1% on MATH500; Dr.GRPO pushed further to 78.2% GSM8K and 45.6% MATH500.

AI Agent Framework from Scratch GitHub **2025**

- Built a **multi-step tool-use** agent from scratch in Python – full think, tool, observe loop with OpenAI function calling, Pydantic tool schemas, and a callback-driven executor for web search, sandboxed code execution, and file I/O.
- Integrated Model Context Protocol (**MCP**) so external tool servers plug in at runtime without custom per-tool wiring – decouples the agent core from tool implementations.
- Designed sliding-window memory with **automatic compaction** for long sessions; exposed the agent via a FastAPI backend with streaming responses, session persistence, and queryable tool-execution logs.
- Evaluated on the **GAIA benchmark** (multi-step real-world tasks: web search, documents, arithmetic, code); compared GPT-4o vs Claude tool-use strategies to refine system prompts and dispatch logic.

Coding-Agent GitHub **2026**

- Built a **LangGraph CLI** coding agent with file read/write, grep, directory listing, and shell tools; deployed inference on vLLM with automatic git-root detection so it runs on any local repo.
- Integrated LangSmith tracing across the **plan/execute graph** and per-step LLM calls; simplified the codebase by removing eval harnesses, metrics, and routing while keeping end-to-end repo editing.

Mini-Inference-engine GitHub **2026**

- Built a single-GPU LLM inference engine for Qwen2.5-7B-Instruct with paged KV cache and **custom PagedAttention**, using a shared 512x16 token block pool (8K tokens) instead of reserving max context per request.
- Designed **continuous batching** with FCFS scheduling on H100 80GB (8 concurrent 128-token decodes): 139.8 vs 53.6 tok/s (2.6x throughput), 7.3s vs 19.1s wall time (2.6x), with multiple requests sharing each forward pass.
- Benchmarked Paged KV vs max-length reservation on a 32K-token budget: 226 vs 64 concurrent sequences (3.5x, 93% less fragmentation).
- Evaluated TTFT on H100 80GB under a 16-request burst: max TTFT fell from 35.5s to 0.22s (160x) vs one-at-a-time scheduling.

Experience

Atrium (Pfizer Engagement) **Jun 2025 - Aug 2025**

AI Engineer *Silver Spring, USA*

- Designed and built an end-to-end document generation system for Pfizer, parsing clinical-trial protocols into a Neo4j **knowledge graph** and creating a text-to-Cypher retrieval pipeline with **LangChain**.
- Built a custom **LLM-as-a-Judge** on faithfulness, correctness, and hallucination risk as the prompt-tuning feedback loop, and used it to improve consistency by extracting stylistic patterns from prior SAPs as injected context.
- Cut first-draft generation time from a traditional **1–2 days to 1.0–3.4** minutes across 71 SAPs, with the work **co-authored as a peer-reviewed publication** in Clinical Trials (SAGE).

Tezo **Jul 2021 - Jul 2024**

Software Developer (ML) *India*

- Built a **RAG-powered chatbot** over internal SharePoint documents using LangChain, OpenAI text-embedding-3 embeddings, and a Chroma vector store with structure-aware chunking; deployed to 200+ employees, unifying search across 1,000+ documents and cutting document lookup time by 61%.
- Improved fraud detection by replacing a logistic regression baseline with **XGBoost**, boosting recall by **12%**.
- Engineered a data pipeline ingesting source data via SnapLogic into **Snowflake**, writing **stored procedures** and maintaining a three-tier dev/stage/prod schema architecture for systematic, controlled promotion of data to production.

Technical Skills

Languages & Frameworks: Python, PyTorch, Triton, SQL, Pydantic, FastAPI, LangChain, LlamaIndex
ML & Training: LoRA/QLoRA, FSDP, Hugging Face Transformers, Unsloth, W&B, Megatron, DeepSpeed
Infrastructure: Docker, Kubernetes, AWS SageMaker, Azure ML, Snowflake, GitHub Actions, CI/CD

Education

University of Maryland, College Park **May 2026**

MS, Applied Machine Learning *College Park*

- **Coursework:** Deep Learning, NLP, Advanced ML, Reinforcement Learning, Optimization, Probability & Statistics