Akhil Shekkari

+1 (425) 426-8292 | shekkari.akhil@gmail.com | linkedin | github | portfolio

SUMMARY

AI Engineer with **3+** years of industry experience building and deploying production LLM systems. Specialized in Agents, large scale training, and inference optimization, with a focus on evaluation and scalable system design.

EXPERIENCE

Atrium **Jun 2025 - Aug 2025**

AI Engineer *Silver Spring, USA*

- **Co-authored a peer-reviewed publication** in Clinical Trials (SAGE) titled Leveraging Generative AI to Transform Statistical Analysis Plan Authoring in Clinical Trials
- **Automated Statistical Analysis Plan** generation from clinical trial documents for **Pfizer's** team, reducing drafting time by **60%.**
- **Evaluated knowledge graphs**, naive and semantic chunking strategies; benchmarked embedding models, rerankers, and HyDE then selected a RAG configuration that improved retrieval relevance by **18%** over baseline.
- **Built a custom LLM-as-a-Judge** evaluation system with structured rubrics scoring completeness, guideline adherence, and hallucination detection against ground truth SAPs, achieving **82%** precision and **78%** recall.
- **Conducted multiple ablations** improving generation consistency by **25%** and reducing manual revision cycles by **33%.**

Tezo **Jul 2021 - Jul 2024**

Machine Learning Engineer *India*

- **Designed a RAG powered chatbot** over SharePoint repositories, giving **200+** employees a unified search interface across **1,000+ internal documents** with sub-second retrieval.
- **Engineered semantic search** pipeline (embeddings + vector DB) reducing document lookup time by **61%**, and added LLM-based summarization cutting manager review time by **37%**.
- **Optimized chunking and retrieval** by benchmarking chunk sizes, overlap strategies, and hybrid search, improving answer relevance and reducing hallucination rates across document types.
- **Worked on evaluation pipeline** to measure RAG output quality, tracking faithfulness, retrieval relevance, and hallucination rates across queries, enabling iteration on the retrieval and generation stages based on measured metrics
- **Trained ML models** for insurance policy scoring and fraud detection, improving **fraud recall by 12%** and enabling earlier intervention across the claims pipeline.

PROJECTS

Distributed Training from first principles | Github **2026**

- **Implemented GPipe and 1F1B** pipeline scheduling on a **multi-GPU** transformer pipeline with stage-wise model partitioning and micro-batch execution for scalable distributed training.
- Built **distributed training infrastructure** using core **NCCL primitives** (send/recv, broadcast, all-reduce) to synchronize activations and gradients across pipeline stages during forward and backward passes.
- **Benchmarked GPipe vs 1F1B scheduling** on a 4-stage transformer pipeline across 4 GPUs; 1F1B improved pipeline **utilization** by **~3%** and reduced **communication overhead** by **~4%, idle bubbles** by **~4%** overlapping forward and backward micro-batches.
- **Extending** the distributed training framework to support **tensor parallelism** alongside pipeline parallelism, targeting hybrid **3D parallel** training architectures.

LLM Inference Engine from Scratch **2026**

- **Built a modular inference engine** with Triton-fused FlashAttention kernels and paged KV-cache, achieving **2.3x** speedup over Python-level implementations on **4K–16K** token workloads.
- Implemented **speculative decoding** using a **Qwen3 - 0.6B** draft model with **Qwen3 - 4B as the target verifier**, reducing autoregressive decoding latency by **~28%** and increasing token throughput by **~1.4×** on long-context workloads.
- **Designed a dynamic batching and request scheduler** to serve concurrent inference requests, improving GPU utilization by **22%** and sustaining **1.6×** higher tokens/sec compared to naive sequential decoding.
- Profiled **GPU kernels** with **Nsight** Compute to identify memory-bound bottlenecks; optimized tiling and memory access patterns to reach **~78%** of peak **HBM** bandwidth on **A100**.

Multi-Agent Reasoning System with GRPO | Github **2025**

- Designed and built a modular **multi-agent grpo system** (Planner, Executor, Verifier, Memory) with tool-grounded reasoning, supporting up to **20 decisions/query (4 rollouts × 5 turns)** and delivering a **+300%** exploration gain over single-trajectory RL.
- Engineered an end-to-end data pipeline to normalize, merge, and parquet-export **DeepMath-103K + FlashRAG-NQ** into a unified **182,190**-sample training corpus with balanced coverage **(56.5% math, 43.5% QA)**.
- Implemented **4-bit QLoRA** fine-tuning for a **1.5B** policy model with **LoRA** , training only **1.10%** of parameters for efficient RL adaptation.
- Achieved strong pilot training gains: average reward improved from **29% to 87%** while loss decreased by **97.7%** in initial GRPO runs.

TECHNICAL SKILLS

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

EDUCATION

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

Master of Science, Applied Machine Learning

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