Shubham Patel

shubham.patel7177@gmail.com | +1 (945) 274-5011 | Github |LinkedIn

EXPERIENCE

Research Assistant **May 2024 – Present**
Center for Applied AI & Machine Learning, The University of Texas at Dallas Richardson, TX

- **End-to-end optimization:** Designed a fine-tuning framework for decoder-only LLMs in Retrieval-Augmented Generation (RAG), enabling end-to-end optimization with gradient flow to the retriever, overcoming the non-differentiability of prompt augmentation.
- **Knowledge Organization**: Developed information-theoretic scoring mechanisms to strategically select documents for the datastore, approximating the intractable optimization of jointly determining the optimal LLM parameters and datastore contents, achieving a 30% improvement in accuracy per unit of memory.
- **Complex System Implementation:** Engineered a comprehensive, modular RAG codebase with HuggingFace integration including retrievers like Dense Passage Retrieval (DPR) with FAISS, advanced document processing (temporal scoring, recency, relevance), comprehensive and efficient training pipelines and robust evaluation (embedding-based similarity) with efficient batched inference.
- **Scale:** Trained LoRA adapters for Llama-3-8B and DPR with mixed precision, 8-bit quantized weights, 8-bit optimizer and gradient checkpointing along with a custom loss function resulting in 70% reduction in memory usage while maintaining same performance.

Research Assistant **Aug 2024 – Nov 2024**
Computer Vision and Multimodal Computing Lab, The University of Texas at Dallas Richardson, TX

- **Architecture Design:** Created a novel cross-attention mechanism aligning temporal audio features with spatial embeddings for active speaker detection, successfully implementing and training a complete end-to-end model on AVA Active Speaker dataset.
- **Model Implementation:** Engineered a transformer-based multimodal architecture by implementing dataloaders with frame-audio alignment, convolutional audio encoders, multi-head self-attention for visual and audio features, MLP-based fusion layer, dense prediction head and evaluation logic.
- **Annotation:** Developed an automated annotation pipeline and Python-based labeling tool by implementing multi-face detection and tracking algorithms and an efficient UI design, reducing annotation time by 75%.

Student Researcher **Jan 2024 – May 2024**
Computer Vision and Multimodal Computing Lab, The University of Texas at Dallas Richardson, TX

- **Two-Stage Fine-Tuning:** Created a two-stage fine-tuning method for diffusion models by implementing unique identifier token learning from single images, enabling personalized image editing with strong perceptual similarity (average LPIPS of 0.2) and high semantic compatibility (average CLIP-T score of 29) between generated outputs and source images.

EDUCATION

The University of Texas at Dallas **Aug 2023 – May 2025 (expected)**
Master of Science, Computer Science | Intelligent Systems track (GPA: 4.0/4.0) Richardson, TX
Indian Institute of Technology Madras **Jul 2019 – May 2023**
Bachelor of Technology, Mechanical Engineering | Minor in AI and Machine Learning Chennai, India

SKILLS

- **Core ML/AI:** LLMs, Retrieval-Augmented Generation, Model Alignment, Multimodal Learning, Diffusion Models
- **Frameworks & Libraries:** PyTorch, HuggingFace, transformers, OpenCV, scikit-learn, Weights & Biases
- **Languages & Dev Tools:** Python, C++, SQL, Java, Git, CUDA, React, Node.js
- **Infrastructure:** AWS, Docker, Ray, Kubernetes, MLflow, Apache Spark

PROJECTS

AI Systems | Technologies: PyTorch, HuggingFace, Ray, AWS, Docker, MLflow

- **Reinforcement Learning for LLM Alignment:** Implemented a distributed PPO pipeline with Ray to fine-tune Pythia-1.4B using RLHF on Stanford's Human Preferences dataset. Reduced harmful outputs by 23% and increased helpfulness by 18% with semantic similarity feedback.
- **Model Deployment Pipeline:** Created a Docker-based inference service on AWS ECS for HuggingFace models with MLflow experiment tracking. Reduced deployment time by 37% and inference latency by 28% versus manual workflows.