

Kaushik Dwivedi

dwivedi.kaushik24@gmail.com | +91 79855 35913 | github.com/kaushikd24 | New Delhi, India

EDUCATION

Birla Institute of Technology and Science, Pilani (BITS Pilani), India

M.Sc. Mathematics and B.E. Mechanical Engineering (Dual Degree)

Oct 2020 – May 2025

SKILLS

Programming Languages: Python, C++, Java, Bash, SQL, MATLAB

ML Frameworks: PyTorch, TensorFlow, Keras, Scikit-learn, JAX, HuggingFace Transformers, OpenCV, Pandas, PySpark

MLOps Tools: Docker, FastAPI, Flask, GCP (Compute Engine, Cloud Storage, Cloud Run), Kubernetes, Linux, Apache Spark, CI/CD (GitHub Actions, Jenkins), LangChain, LlamaIndex, LangGraph, AutoGen

Coursework: NLP, DSA, OOP, Deep Learning, Computer Vision, System Design, Reinforcement Learning

Machine Learning Engineering Intern

Syna Group, New Delhi

June 2024 – April 2025

- Fine-tuned **Mistral-7B** and **LLaMa-2-7B** using **LoRA/QLoRA** for Q&A and summarization over mining and land acquisition documents; enabled 40% faster compliance audits, communicated key insights to team leads and senior managers.
- Developed **Retrieval-Augmented Generation (RAG)** pipelines using **LangChain, LlamaIndex, and FAISS**, and chunked document embeddings to build multi-hop enterprise agents; improved search relevance by 35%.
- Prototyped agentic workflows with **LangGraph + AutoGen** to create multi-step LLM agents capable of dynamic tool use, long-context document interaction, and chain-of-thought reasoning.
- Deployed scalable inference services using **FastAPI + Docker + GCP** with integrated **LangChain server and tracing**; achieved 21% latency reduction via multithreaded inference tuning.
- Applied **Temporal Fusion Transformers (TFT)** for forecasting time series and asset demand in real estate portfolios; achieved 82% accuracy, enabling proactive asset allocation.
- Conducted internal benchmarking on LLaMA and Mistral models with LangGraph and AutoGen for agentic task routing and dynamic tool use workflows.

Undergraduate Researcher – Deep Learning & Computer Vision

BITS Pilani, India

GitHub

Aug 2023 – May 2024

- Benchmarked **CNNs, ResNets, ViTs (Vision Transformers)**, and **MLP-Mixers** for image classification and detection tasks, focusing on trade-offs between data size and model parameters.
- Designed **ViT-MLP** hybrid models optimized for embedded deployment, reducing inference time by **30%** with **<2%** drop in accuracy. Key result: Fixing **attention sequences** to less than 197 enable ViTs to work on less memory.
- Achieved **82% Top-1 accuracy** on CIFAR-10 using a custom PyTorch implementation of the **MLP-Mixer** architecture.
- Investigated robustness of ViTs and MLP-Mixers to shuffled patch orders and Gaussian noise. ViTs retained **12% higher accuracy** under moderate perturbations for ImageNet datasets, but **ResNets** work best for less than 100,000 images.

PROJECTS

Decision Transformer-Based Trading Agent – Reinforcement Learning for Financial Markets

GitHub

- Reframed **offline RL** as a **sequence modeling (NLP)** task using a **Decision Transformer** with linear attention (**Linformer, Reformer architecture**) and LayerNorm tuning, optimized using **FlashAttention** to **reduce memory resources by 25%**.
- Trained on **S&P 500 index data** using **PyTorch, Pandas/PySpark**; the model achieved **85% accuracy** on predicting **BUY/SELL/HOLD** actions. Deployed and served via **Gradio, FastAPI, Docker, and Hugging Face**.
- Demonstrated **policy-free learning** using a Decision Transformer trained on custom **HalfCheetah-v5 Gymnasium (OpenAI Gym) environments**, reaching **85% accuracy**, with experiments tracked using **Weights and Biases**.
- **Deployed** the trained model as a **live demo** via **Hugging Face Spaces** using Gradio + FastAPI; accessible at this link: [dt-huggingface-demo](#).

GRPO and DPO-Based LLM Fine-Tuning – DeepSeek RLHF Implementation

GitHub

- Fine-tuned GPT-2 using **LoRA adapters** on the Anthropic HH dataset for human preference modeling without reward models.
- Implemented custom **GRPO (Group Relative Policy Optimization)** and **DPO (Direct Preference Optimization)** training pipelines using PyTorch and Hugging Face Transformers as backup, eliminating PPO instability.
- Benchmarked GRPO, DPO, and PPO across KL divergence, sample efficiency, and preference-aligned generation accuracy.
- Achieved 87% downstream task accuracy; **GRPO consistently outperformed PPO and DPO** in performance, training stability, and used 20% fewer resources.
- **Published `grporlhf` on PyPI** — a plug-and-play GRPO training package installable via `pip install grporlhf`, with integrated **GitHub CI/CD** for auto-testing and publishing, accessible at this link: [grporlhf](#)

BERT4Rec and DeepFM-Based Recommender Systems

GitHub

- Built sequence-based and feature-based **recommendation systems using BERT4Rec and DeepFM** with high-dimensional embeddings to maximize CTR (Click Through Rate) on PyTorch.
- Achieved 0.89 AUC with DeepFM and 0.62 HR (Hit Rate) with BERT4Rec on MovieLens-1M; evaluated using NDCG, AUC-ROC, and CrossEntropyLoss.
- Employed Deep Neural Networks and Factorization Machines along with **collision-less embedding tables** to handle millions of users without performance loss, and avoid reliance on expertise feature engineering in the recommender system.