Divya Kumari

215-892-5386 | divya.kumari32@gmail.com | linkedin.com/in/divya-kumari-b053a8114 | github.com/divya-kumari32

EDUCATION

University of Pennsylvania Aug 2022 – May 2024
Master of Science in Engineering: Computer and Information Science
Thesis: Optimizing Code Generation with Large Language Models (advised under Prof. Osbert Bastani)
Coursework: Analysis of Algorithms, Operating Systems, Artificial Intelligence, Internet and Web Systems, Big Data Analysis

Thapar Institute of Engineering and Technology Aug 2016 – Jul 2020
Bachelor of Science in Engineering: Computer Engineering

TECHNICAL SKILLS

Languages: Python, C/C++, SQL, JavaScript, TypeScript, Java
Algorithms: GRPO, DPO, PPO, RLHF, SFT, Speculative Decoding

PROFESSIONAL EXPERIENCE

Research Engineer | *Python, PyTorch, FSDP, OpenShift, Coreweave* Jun 2024 – Present
IBM Research *NY, USA*

- Enabled agentic RL rollouts and trajectory generation at scale by building an agentic browser pipeline across 23 real-world environments (Gmail, Slack, Notion, etc.), 120+ tasks each; adapted WebArena-Infinity with a custom agent framework on open-source models to replace proprietary Claude-based agents.
- Delivered IBM's Granite 4 flagship architecture by pretraining a novel hybrid Mamba2 model from scratch on 192 A100 GPUs across ~2.2T tokens, outperforming Llama 3.1 8B on L1/L2 benchmarks despite 7x fewer training tokens. Co-authored the public technical blog on Hugging Face and open-sourced the full training recipe.
- Extended model reasoning to 64K-token contexts by engineering a complete post-training pipeline for a 70B+ Llama model on 512 to 768 H100 GPUs , 4-round iterative SFT with think/no-think training (2:1 ratio), followed by GRPO-based RL across three progressive context phases (8K → 32K → 64K tokens).
- Improved reasoning benchmark scores by 35%+ by building a synthetic data generation loop from the final RL checkpoint: evaluation-driven corpus gap analysis, reasoning trace generation across math, code, and tool-use, with rule-based and reward-model filtering for correctness verification.
- Enabled seamless continued pretraining across training runs by implementing a WSD learning rate schedule on distributed clusters (Vela, CoreWeave, Kubernetes/OpenShift); corpus spanning Nemotron-CC, FineWeb-Edu, DCLM, and EAI.

Graduate Research Assistant | *Python, PyTorch, JavaScript, MongoDB, AWS, Flask* Jul 2023 – Dec 2023
University of Pennsylvania – advised under Prof. Lyle Ungar *PA, USA*

- Reduced inference overhead in interactive NLP applications by investigating speculative decoding techniques for low-latency autoregressive generation, integrated into voice-based educational tools deployed in classroom settings.
- Enabled dynamic multi-party conversation handling by developing context-sensitive dialogue systems with interruption handling and speaker diarization, integrating PyTorch-based models with a Flask/AWS backend.

Machine Learning Research Assistant | *Python, TensorFlow, NumPy, Pandas* May 2023 – Jul 2023
Boston University – advised under Prof. Dokyun Lee *MA, USA*

- Achieved 30% improvement in model accuracy and 10% increase in adaptability by training and fine-tuning open-source LLMs for sociopolitical context analysis using gradient boosting and regularization.

Software Engineer | *TypeScript, Angular, Docker, Kubernetes, Jenkins, Camunda* Aug 2020 – Aug 2022
Cisco Systems *Bangalore, India*

- Drove 30% performance boost, 25% higher client satisfaction, and 50% user retention lift by designing a microservices architecture with scalable APIs and automated device onboarding pipelines; owned end-to-end CI/CD framework that significantly reduced deployment time, improved release cadence, and enhanced team velocity.

PROJECTS

PennSearch – Distributed Search Engine | Java, AWS, Spark Jan 2024 – May 2024

- Built a full-stack distributed search engine comprising a multi-threaded web crawler, inverted indexer, TF-IDF/PageRank ranker, and interactive query UI, enabling scalable full-text web search on AWS infrastructure.

PennOS | Docker, C Jan 2023 – May 2023

- Delivered a fully functional UNIX-like OS with foreground/background process management, priority round-robin scheduling, a FAT file system, and an interactive bash-like shell supporting piped commands.