

Yash Lalit Vanjani

(412) 805-1470 • yvanjani@alumni.cmu.edu • [LinkedIn](#) • [Github](#) • [Google Scholar](#)

Education

Carnegie Mellon University

Master of Science, Machine Learning & Systems (GPA: 3.8/4.0)

Pittsburgh, Pennsylvania

Aug 2021 - Dec 2022

Indian Institute of Technology

Bachelor of Technology, Computer Science & Mathematics

Guwahati, India

Jul 2014 - May 2018

Coursework

Machine Learning

Advanced Natural Language Processing

Intro. to Computer Systems

Deep Learning

Deep Reinforcement Learning

Distributed Systems

Computer Vision

Search Engines

Storage Systems

Publications

- Layer Sharding for Large-Scale Training with Muon ([blog](#)).
- Data Curation Just Got Smarter: How Essential AI Used AMD GPUs to Label the Web for Smarter AI ([blog](#)).
- Rethinking Reflection in Pre-Training ([arXiv paper](#) | [blog](#)).
- Essential-Web v1.0: 24T tokens of organized web data ([arXiv paper](#) | [HF dataset](#)).
- Practical Efficiency of Muon for Pretraining ([arXiv paper](#) | [blog](#)).

Invited Talks

- Large-scale training on AMD GPUs @ AMD

Jul 2024

Skills

- Python, C++ • JAX, PyTorch, CUDA, Triton, Pallas • vLLM, SGLang • Kubernetes, Spark, Grafana

Work Experience

Essential AI - Member of Technical Staff | [Machine Learning Infrastructure - Pre-training](#)

Sep 2024 - Present

Scaling foundation model pre-training on 1000s of AMD GPUs and Google TPUs.

- **Optimized distributed multi-node training performance for dense model sizes upto 32B, MOE model sizes upto 1T, and context lengths upto 128K** by using a combination of parallelization strategies like Data Parallelism (DP), Fully-sharded Data Parallelism (FSDP), Tensor Parallelism (TP), Sequence Parallelism (SP), Context Parallelism (CP), Expert Parallelism (EP), and Pipeline Parallelism (PP).
- **Orchestrated deep-kernel profiling using JAX profiler and optimization** of LLM training workloads, delivering a **25% throughput gain** and **\$3M in annual cost avoidance**.
- Conducted cross-hardware benchmarking of models, evaluating **sharding strategies optimized for hardware-specific communication primitives under differing interconnect topologies** to maximize scaling efficiency.
- **Innovated layer sharding** to improve the performance of SOTA optimizer - Muon by 30x, making it viable for large-scale LLM pre-training ([blog](#)).
- **Devised innovative document masking among packed documents in JAX**, increasing performance on evals by upto 10%, reducing the LLM pre-training time by 25%, and **\$6M/year in cost savings**.
- **Pioneered novel attention algorithms in fused flash attention kernel**, allowing rapid experimentation.
- **Architected a distributed inference system** in Kubernetes that scaled horizontally up to the entire fleet of GPUs, sustaining 1M concurrent requests/s, and led to scaling synthetic data curation jobs seamlessly.
- **Built a KV cache-aware load balancer for large-scale inference system** that curtailed tail-latency by **40%** and trimmed infrastructure spend by **\$2M annually**.
- **Diagnosed and eliminated inference performance choke points** across MoE models like DeepSeek-V3, and dense models like Llama3 and Qwen 2.5 through rigorous benchmarking.
- **Enabled state-of-the-art inference optimizations** (prefix caching, chunked prefill), elevating generation throughput by **30%**.
- **Authored a fault-tolerant inference client SDK** capable of sustaining **1M concurrent streams**, saturating a 1,024-GPU AMD fleet that scaled to a throughput of **250M tokens/s**.
- **Directed large-scale training dataset curation** using advanced extraction, de-duplication, and semantic annotation - raising training efficiency and downstream task accuracy by 10%.
- **Fully automated data-pipeline orchestration** via Airflow and Terraform, shrinking manual touchpoints by 90% while ensuring immutable, audit-ready workflows.
- *Tech stack: Python, C++, JAX, Triton, Pallas, MaxText, vLLM, SGLang, Spark, Grafana, Airflow, GCP.*

- Designed on-demand **LSM-memtable checkpointing**, enabling rapid crash recovery and durable persistence with near-zero write-path overhead.
- **Instituted a cross-layer “hung-disk” detection framework** - correlating filesystem, kernel, and Stargate telemetry—to preempt catastrophic stalls in dense storage clusters, **averting 24 to 48 hour outages** per incident.
- Root-caused and fixed **race conditions** in the distributed-storage core, cutting fleet-wide service crashes by 10%.
- Deployed an **on-device Retrieval Augmented Generation (RAG) - powered JIRA bot** that auto-summarizes ticket threads, boosting engineering situational awareness and reducing daily triage time.
- Crafted a **fine-grained product-versioning framework**, simplifying feature gating and reducing backport effort.
- *Tech stack*: C++, RocksDB, ZooKeeper, Cassandra, Protocol Buffers, Ollama.

- **Owned and modernized key subsystems** -publishing server, online review, update-notification, and logging.
- Optimized the PDF→XML comments-import algorithm, tripling throughput and improving data-conversion fidelity.
- Integrated AEM version-control APIs, unifying content lifecycle management and lowering publishing errors.
- Hardened middleware design and eliminated deep memory leaks, driving a 100× increase in server availability.
- Architected Document Cloud online review workflows in FrameMaker, enabling real-time **multi-author collaboration** for tens of thousands of technical-documentation users.
- Pioneered FrameMaker’s first React-based feature, the Comments Review Panel, instituting a novel front-end style.
- Devised an HTML5 nested-list dynamic-filtering algorithm that streamlined content navigation in published docs.
- *Technologies used*: C++, **Node.js**, Express.js, MongoDB, React.js.

Research Experience

Investigations of Transfer Learning

Feb 2022 - May 2022

Dr. Bhiksha Raj, Language Technologies Institute, School of Computer Science, CMU - [report](#)

- Hypothesized and demonstrated use of **larger representation size** leads to generation of **better generalizable features** in Transfer learning.
- Experimented with final representation size of various model families of ResNet and EfficientNet on **Classification tasks** through Transfer Learning on the PASCAL VOC2007 dataset.

Projects

PyTorch like Deep Learning framework implementation from scratch

Feb 2022 - May 2022

- Linear layer, Convolutional layer, Pooling, Down-sampling, Up-sampling, Batch Normalization, Dropout.
- RNN Cell, GRU Cell, CTC Loss, Greedy Search, Beam Search, Weight Tying and Decay, Activity Regularization.
- Automatic Differentiation, Computation Graph, NDArray implementation with CUDA backend.

Deep Learning based NLP & CV tasks using PyTorch

Feb 2022 - Dec 2022

- NLP system from scratch for Scientific **Named Entity Recognition** with **HuggingFace Transformers** ([link](#)).
- Mini version of **BERT** for Sentence Classification on CFIMDB & STT dataset ([link](#)).
- End-to-End Speech to text transcription by using **Attention** and Pyramidal BiLSTM.
- Automatic Speech Recognition: ConvNext to generate embeddings & bi-directional LSTM for phoneme prediction.
- Face Classification & Verification using CNNs: Implemented ResNet, MobileNet, and ConvNext.

Reinforcement Learning algorithms in OpenAI Gym

Sep 2022 - Dec 2022

- Policy & Value Iteration, Policy Gradient algorithms, Imitation Learning: Behavior Cloning & DAGGER.
- Deep Q-learning, Model-based RL, MuZero.

Text Search Engine

Sep 2022 - Dec 2022

- Implemented Information Retrieval algorithms BM25 & Indri with **Diversity** algorithms: PM2 & xQuAD.
- Supported **Query Expansion**, Query Multi-representation and Sequential Dependency Models.
- Boosted performance by **Re-ranking** results by availing - SVM rank, Coordinate Ascent, and ListNet algorithms.

Distributed Systems

Feb 2022 - May 2022

- Created a Remote Method Invocation (RMI) library.
- Implemented RAFT Consensus Protocol for **Consistency and Partition Tolerance** in a Distributed System.

Cloud-backed Local FileSystem — [report](#)

Nov 2021 - Dec 2021

- Created a FileSystem to store small files and all metadata locally, and large files in cloud.
- Accomplished block level **de-duplication** of files stored in cloud to decrease cloud capacity consumed.
- Added **Cache** layer to speed up all cloud operations, and supported backup facility via **Snapshots**.

Awards & Achievements

- Awarded **\$10K scholarship** from the Information Networking Institute for graduate studies at CMU.
- Secured **All India Rank 1875** in **IIT-JEE Advanced** 2014 among **1.4 million** participants.