Akshay Umesh

+1-862-423-9778 akshayumesh2000@gmail.com https://www.linkedin.com/in/akshayumesh/ https://github.com/Akshay879

Skills

- **Programming Languages**: Python, Java, Javascript
- **Frameworks**: PyTorch, Langchain, Expressjs, Nodejs
- **Core Skills**: Machine Learning, Deep Learning, Generative AI, Cross-Modal/Multimodal Generative Modelling, Computer Vision, Natural Language Processing, AWS, RAG, Autoregressive Modelling, SQL

Education

New Jersey Institute of Technology **Sep 2023 - May 2025**
M.S., Computer Science

JSS Academy Of Technical Education **Aug 2017 - Jul 2021**
B.E., Computer Science Engineering

Research

Micro-Budget Latent Diffusion Transformer for High-Resolution Image Synthesis Jan 2025 - Mar 2025

- Developed a **1.2B-param Latent Diffusion Transformer** (PyTorch) from scratch, achieving a **12.7 FID** on the COCO dataset at a **$1314** compute cost; trained for **54.96 8×H100 hours** on **25M image-caption pairs (SA1B, JourneyDB, DiffusionDB).**
- Engineered a progressive **pretraining-finetuning pipeline**, seamlessly scaling the model's denoising capabilities across resolutions by dynamically adjusting positional embeddings, enabling effective **masked and unmasked denoising representation learning.**
- Optimized pretraining by implementing a **75% patch masking ratio**, minimizing input sequence length, and integrating a **lightweight patchmixer** to retain crucial global semantic information, leading to superior representation learning.
- Implemented a **ResNet-style Diffusion Transformer**, integrating a **6-layer patchmixer and 28-layer DiT backbone**, which leverages **cross-attention on OpenCLIP text embeddings**, noise-modulated information flow, and **Expert Choice Mixture of Experts** within a VAE latent space.
- Modeled the diffusion process as a **Stochastic Differential Equation** (SDE), as explained in **score-based generative models** and denoising score matching.
- Scaled model training efficiently using **Fully Sharded Data Parallel (FSDP)** and the **Composer Hydra framework**.

Projects

Transformer based Autoregressive Image generation using VQGAN Nov 2024 - Jan 2025

- Developed a generative model in PyTorch from scratch, combining a **108M-param VQGAN (vocab size: 8192)** for discrete neural representation learning and a **91.5M-param GPT-2 (257-token context)** for autoregressive image generation; implemented a **VQ-VAE** with adversarial training using a **ResNet-based encoder-decoder, self-attention, and multi-objective loss (reconstruction, perceptual, adversarial, quantization, commitment)** to learn structured latent codes.
- Designed quantization with **gradient flow reparameterization** for effective backprop through discrete latents; built a pre-layer norm GPT-2 with **4-headed causal self-attention** to model chained categorical posteriors; trained on the **Oxford Flowers dataset (8189 images) in BFloat16 mixed precision** for high-fidelity synthesis.

Foundation Model for Text Completion Sep 2024 - Nov 2024

- Implemented a GPT-2-based autoregressive language model with **124M trainable parameters and a 1024-token context length** from scratch.
- Trained on **10B tokens (FineWebEdu dataset)**, achieving a validation loss of **3.1440** and HellaSwag score of **30.22%,** outperforming **OpenAI-GPT2 124M (trained on 100B tokens)** in learning efficiency (~10x).
- Scaling to **50B** tokens matched OpenAI-GPT3 124M's HellaSwag score of **33.70%.**
- Optimized training on an 8x A100 SXM cluster (12 hours, 50B tokens) using **PyTorch DDP, BFloat16 mixed precision, flash attention, and kernel fusion.**

U-Net for Image Reconstruction **Dec 2024 - Jan 2025**

- Implemented a U-Net architecture in PyTorch from scratch, incorporating a **4-stage convolutional encoder (64->512 channels) and a 4-stage decoder (512->64 channels)** for high-fidelity image reconstruction.
- Designed an encoder with multi-stage downsampling, self-attention (1024 latent channels), and residual feed-forward layers, optimizing for 16x compression in latent space (256x256 ->16x16 feature maps).
- Developed a decoder using **transposed convolutions (2x upsampling per stage) and skip connections**, improving feature retention and generating realistic RGB images (256x256, 3 channels).
- Applied custom **Kaiming initialization** for weight scaling, reducing initialization variance and stabilizing training.
- Trained using **BFloat16** mixed precision, achieving a **50% memory footprint reduction**, enabling larger batch sizes on A100 GPUs.

Work Experience

Accenture **Sep 2021 - Aug 2023**

QA Team-Kaiser Permanente

- Worked in the QA team for KP-HIE using **Selenium**, **Jira**, and **Jenkins** framework, improving testing processes and ensuring project milestones were met on time
- Embraced Agile methodologies to expedite project delivery, slashing timelines by **30%** and boosting team adaptability.
- Enhanced testing efficiency and accuracy for Kaiser's **Genomics** and **NIPT** projects, leading to more reliable test outcomes and improved project quality
- Ensured adherence to quality assurance and testing standards, resulting in the delivery of reliable and robust software solutions