

Nirmalya Mallick Thakur

+91 77973 95326 | nirmalya23@iiserb.ac.in | [LinkedIn](#) | [Github](#)

EDUCATION

Indian Institute of Science Education and Research, Bhopal

2023 – Present

Bachelor of Science in Computer Science

TECHNICAL SKILLS AND INTERESTS

Programming Languages: Python, C, C++, Java Script

Areas of Interest: LLMs, Transformers, Neural Audio Codecs, Computer Vision, Graphics

Programming, Simulation

Coursework: Econometrics, Linear Algebra, Calculus of one variable, Introduction to Programming in C

Libraries: Pytorch, Pandas, NumPy, Matplotlib, Librosa, OpenGL, DearImGUI

PROJECTS

ENCODEC | *High Fidelity Neural Audio Compression (Paper Reproduction)* [Github Link](#)

- Tools and technologies used: PyTorch, Google Colab, Librosa
- Reproduced a simple version of Meta AI's Encodec paper from scratch inspired by their codebase
- Created a convolution based encoder-decoder architecture with residual vector quantization (RVQ) for compressing latent audio embeddings into discrete codes
- Implemented a multi-scale STFT discriminator with adversarial loss to enhance output quality
- Trained the model using several loss components like reconstruction loss, perceptual loss, commitment loss and generator loss on the LibriSpeech development dataset
- The model can handle multiple target bandwidths like 1.5, 3, 6, 12, 24 kbps and output in both streaming and non-streaming fashion

DeepLLMs | *BPE Tokenizer and Decoder-only Transformer from scratch* [Github Link](#)

- Tools and technologies used: PyTorch, Google Colab
- Implemented a custom Byte Pair Encoding(BPE) tokenizer from scratch and trained it on 100K characters achieving a compression ratio of 3.61x
- Developed a 13M-parameter decoder-only Transformer model inspired by the paper "Attention is All You Need"
- Implemented Multihead Self Attention from scratch with 6 single heads
- Trained the model using the custom tokenizer on two datasets of roughly 500K and 1.5M tokens and generated human like outputs

Sim3D | *Physics Engine for real-time Cloth and particle simulation* [Github Link](#)

- Tools and technologies used: C++, OpenGL, DearImGUI, GLSL(OpenGL Shading Language)
- Developed a 3D physics engine written entirely from scratch in C++ using OpenGL to simulate Newtonian Mechanics, Particle-particle interactions, and real-time cloth simulation using mass-spring systems
- Particle system with efficient collision detection using Uniform grid partitions
- Interactive scene editor using ImGui for easy configuration and gizmos for intuitive scene manipulation
- Applied OOP concepts to abstract and integrate different engine components, improving engine design

U-Net based diffusion model for image generation [Github Link](#)

- Tools and technologies used: PyTorch, Google Colab
- Created a simple U-net architecture with a Linear noise scheduler based on the original "DDPM" paper
- Incorporated multiple upSampling, downSampling, residual and attention blocks into the model
- Experimented with multiple datasets for training the model and got good results on MNIST dataset

POSITIONS OF RESPONSIBILITY

Core Committee: CyberHeathens, Coding Club, IISER Bhopal

2023 – 2024