Swayam Singh

singhswayam008@gmail.com +91 9116277756 GitHub LinkedIn Google Scholar

Education

University of Allahabad 2020 – 2024
B.Tech, Computer Science and Engineering CGPA: 9.56

Experience

Microsoft Research 10/2024 – Present
Research Fellow
- Engineered an end-to-end code autoformalization pipeline in Rust, translating natural-language and informal specs into formally verifiable representations.
- Developing foundational algorithms that exploit reinforcement-learning gradient-optimization dynamics to improve representation efficiency, stability, and generalization.
- Developed an LLM on par with GPT-4o for code generation, instruction-following, and reasoning; post-pretraining and online/offline RL.
- Developed SeleKT, a supervised fine-tuning method that preserves generalization while significantly improving downstream performance.
- Led large-scale training on multi-GPU clusters, optimizing distributed workflows for high-performance code synthesis.

NumPy 12/2024 – Present
Maintainer
- Review, merge, and evaluate core PRs affecting dtype architecture, numerical semantics, and cross-platform behavior.
- Maintain core NumPy with a focus on dtype infrastructure, numerical correctness, and long-term maintenance.
- Work with contributors across architectures on portability, performance, and ABI concerns.

Quansight 07/2024 – 09/2024
Open Source Intern – Numerical Types & Precision (NumPy OSS)
- Led development of NumPy-QuadDType, a cross-platform 128-bit floating-point system for consistent quad precision beyond platform-dependent long double.
- Implemented full dtype ecosystem integration: casting, ufunc dispatch, scalar types, memory model, and serialization.
- Built multi-platform distribution and testing infrastructure handling compiler toolchains, SIMD availability, FMA support, and architecture-specific behavior.

dataX.ai 05/2022 – 10/2022
Machine Learning Engineer Intern
- Built deep-learning models for vision and language; created an ONNX conversion API with Triton deployment, cutting VM load by 12%.
- Wrote custom CUDA kernels for 3D medical-scan processing, achieving a 2x speedup in segmentation over existing GPU solutions.

Publications

- **NextCoder: Robust Adaptation of Code LMs to Diverse Code Edits** (ICML 2025; ICLR 2025 DL4Code). **Swayam Singh**, Tushar Aggarwal, et al. A synthetic-data generation pipeline and the SeleKT adaptation algorithm that make code LLMs robust to diverse, real-world code edits.
- **Narrow Transformer: StarCoder-Based Java-LM For Desktop** (arXiv:2407.03941, 2024). Kamalkumar Rathinasamy, . . . , **Swayam Singh**, et al. A compact, Java-specialized code language model designed to run efficiently on desktop hardware.
- **OctoPack: Instruction Tuning Code Large Language Models** (ICLR 2024 Spotlight; NeurIPS 2023 Instruction Workshop). Niklas Muennighoff, . . . , **Swayam Singh**, et al. Instruction tuning of code models using natural-language Git commits (CommitPack / CommitPackFT).
- **StarCoder: May the Source Be With You!** (TMLR 2023). Raymond Li, . . . , **Swayam Singh**, et al. (BigCode). A 15.5B-parameter open code LLM trained on 1T tokens of permissively licensed code.

Projects

- **Virtual Clothing Assistant (PyTorch)**: An end-to-end virtual try-on system: ResNet101 and UNet for garment/body segmentation, OpenPose for pose estimation, and a PyTorch VITON pipeline for realistic garment warping. Reaches an SSIM of 0.895 and has 500+ GitHub stars.
- **Numpy-QuadDType (C, C++, Python)**: A cross-platform 128-bit (quadruple-precision) floating-point dtype for NumPy with 100k+ downloads. A portable alternative to long double with full casting, ufunc dispatch, scalar types, and serialization that behaves consistently across compilers and architectures.
- **Bare-Bones Inference (CUDA, C++)**: A minimal inference stack with engine-grade features such as batching and scheduling, but serving a single fused megakernel instead of a conventional multi-kernel pipeline. Built DeepSeek-V2 (236B)'s multi-GPU, intra-node megakernel for NVIDIA B200 (sm_100) GPUs.
- **QBLAS (C, C++)**: A high-performance BLAS library for IEEE-754 binary128 (quadruple) precision, implementing optimized linear-algebra kernels that bring quad-precision numerics to workloads unsupported by standard double-precision libraries.
- **cpp-verify (C++, LLVM, SMT)**: Extends C++ with first-class formal-verification constructs (pre/post-conditions and invariants), lowering specifications through an LLVM-based pipeline and discharging the resulting proof obligations to SMT solvers.
- **MIRA – Multimodal Image Reconstruction with Attention (PyTorch)**: A transformer-based architecture for single-view text/image-to-3D reconstruction using ViT encoders and triplane decoders with cross-attention. Generates 3D mesh and video in under 10s on an A100, with SDXL-integrated diffusion for end-to-end scene synthesis.

Honors and Awards

- **2024** — **Kaggle Competition Expert**: Bronze medal (top 7%) in the UBC-OCEAN competition; top 3% in the *30 Days of ML* challenge.
- **2024** — Invited to **Google Research Week**, Google Research's gathering of AI researchers (keynote by Jeff Dean; sessions on differential privacy, responsible AI, and more).
- **2024** — **OctoPack** accepted as a **Spotlight (top 5%)** at ICLR 2024.
- **2023** — Selected for the **Amazon ML Summer School 2023**.
- **2023** — **Clothes Virtual Try-On** crossed 500+ GitHub stars.

Invited Talks

- **MAMBA: Zero to Hero**, invited talk on State Space Models at Cohere for AI.
- **Provably-Correct Code and Efficient Sparse Training of LLMs**, internal research talk at Microsoft Research.
- **Foundations of Machine Learning**, a GDG On-Campus session on the ML landscape, tooling ecosystem, and emerging directions.
- **From Deep Learning to Large Language Models**, a talk on the foundations of modern AI: deep learning, generative models, and LLMs.

Skills Summary

- **Languages**: C, C++, CUDA, Python
- **ML & Training**: LLM post-training, reinforcement learning (online/offline RL), supervised fine-tuning, multi-GPU distributed training, code-generation models
- **Systems & GPU**: CUDA kernels and megakernels, high-performance numerical computing, IEEE-754 quad precision, SIMD/FMA, performance optimization
- **Tools & Frameworks**: PyTorch, NumPy (C-API internals), LLVM, SMT solvers
- **Open Source**: Core NumPy maintenance, cross-platform C-extension packaging, code review