

Subhadeep Chatterjee

858-346-3765 | suchatterjee@ucsd.edu | linkedin.com/in/subhadeep19 | github.com/suga-ucsd | suga-ucsd.github.io

EDUCATION

M.S. in Electrical and Computer Engineering

GPA: 3.5 out of 4.0 — Coursework: Statistical Learning, Visual Learning, Linear Algebra

University of California, San Diego

Sep. 2023 – May 2025

Bachelors in Electrical Engineering

GPA: 3.6 out of 4.0 — Coursework: Machine Learning, Data Structures and Algorithms

Indian Institute of Technology, Ropar

Jul. 2019 – Mar 2023

TECHNICAL SKILLS

Languages: Python, C/C++, Assembly

Parallel & Distributed Computing: CUDA, CuPy, JAX, PyTorch Distributed, Ray, MPI, NCCL, torch.multiprocessing

Developer Tools: Git, Docker, Linux, Emacs, Vim

Libraries: PyTorch, TensorFlow, LangChain, StableBaselines

Skills: Parallel Algorithms, Distributed Systems, High-Performance Computing (HPC)

EXPERIENCE

Robot Learning Researcher – Existential Robotics Lab (Prof. Nikolay Atanasov)

Apr 2024 – Present

UC San Diego

San Diego, CA

- Led a team of 5 researchers for the development of a **scalable parallel learning framework** for vision-based robotic grasping across 64-128 simulated environments, enabling efficient experimentation at scale.
- Engineered a **multi-process PyTorch pipeline** leveraging Ray and `torch.multiprocessing` to vectorize CNN rollouts, offloading computation to GPUs via `CuPy`, which accelerated SAC training by over 13x and reduced iteration time by 2x.
- Integrated **signed distance function (SDF)** models with JAX to compute auto-vectorized gradients, improving simulation fidelity and enabling differentiable policy optimization for complex robotic tasks.
- Containerized the full multi-GPU workflow using Docker, supporting reproducible, portable training of **transformer-based robotic policies** at scale, facilitating rapid benchmarking and deployment of new experiments for the team.

LLM Agent Inference Engineer – MURO Lab (Prof. Sonia Martinez)

Oct 2024 – Present

UC San Diego

San Diego, CA

- Led the development of a **distributed RAG chatbot** for the MURO lab codebase, enabling new members to quickly understand individual project modules and navigate the full codebase to implement and explore other projects efficiently.
- Implemented Ray Serve-based microservices to parallelize embedding computation and retrieval across CPUs and GPUs, while fine-tuning **Gemma 2.0 9B** using pipeline parallelism and gradient checkpointing, reducing training time and resource usage.
- Optimized dense retrieval workflows with multiprocessing queues, custom CUDA kernels, and memory-augmented RAG pipelines, benchmarking performance against state-of-the-art frameworks like vLLM and TGI to ensure maximum throughput.
- Delivered a high-throughput, production-ready **distributed RAG system** achieving sub-second query latency, demonstrating end-to-end scalability and reliability for large-scale conversational AI.

Machine learning Researcher – ISNL Lab (Prof. Gert Cauwenberghs)

Jan 2024 – May 2024

UC San Diego

20 hrs/week

- Developed and trained an **LSTM model** for brain-intention detection using EEG data from a control group, enabling accurate prediction of neural intentions.
- Implemented **FFT-based pipelines** accelerated with CuPy to efficiently extract intention-related peaks from high-volume EEG signals, reducing computational bottlenecks.
- Scaled **500GB EEG dataset** processing with asynchronous batching, cutting preprocessing time and supporting rapid iterative model training.
- Parallelized the EEG decoding workflow with multithreaded CPU preprocessing and full **GPU time-series inference**, achieving low prediction error and high-throughput performance.

Data Analyst Intern – Samsung R&D

Jun 2022 – Sept 2022

Bangalore, India

40 hrs/week

- Contributed to big data pipelines for the **Bixby Voice Assistant**, enhancing user experience and improving response quality at scale.
- Parallelized sentiment inference using **Ray** batch predictions across CPU cores, reducing response latency from 0.3ms to 0.1ms.
- Optimized speech-to-text analytics with multiprocessing and `CuPy` acceleration, decreasing end-to-end pipeline latency for large-scale audio data.
- Collaborated within a four-member team to integrate scalable solutions, increasing throughput and efficiency of audio processing workflows.

- Developed a deep-learning pipeline for **real-time radio signal classification**, targeting deployment on resource-constrained hardware.
- Implemented GPU-accelerated training in **TensorFlow**, leveraging optimized kernels to achieve faster convergence on large RF datasets.
- Streamed GNURadio signal data with multiprocessing to support continuous RF collection and seamless integration with online classifiers.
- Delivered a proof-of-concept demonstrating low-latency, GPU-based RF classification in real-world scenarios, validating practical deployment feasibility.

PROJECTS

Large-Scale Vision-Language Model Training | *PyTorch, Ray, CUDA, Docker, NCCL*

- Designed and trained a CLIP-style model on **20M image–text pairs** using **PyTorch Distributed (DDP)** with NCCL collectives for synchronized multi-GPU pretraining.
- Engineered a **Ray-based distributed data pipeline** for augmentation + sharded loading, reducing GPU idle time by **35%**.
- Implemented **mixed precision (AMP)**, fused kernels, and tuned inter-GPU communication to achieve **>90% scaling efficiency**.
- Containerized end-to-end training with **Ray + Docker**, enabling reproducible multi-node experiments with fault tolerance.

LLM Inference Engine From Scratch | *C++, Python, gRPC, CUDA*

- Built a GPT-style inference engine with **KV-cache**, **paged attention**, and **incremental token decoding** using contiguous memory layouts.
- Implemented **dynamic batching** and a low-latency **FastAPI/gRPC** server for streaming token-by-token generation.
- Added support for **INT8/FP8** quantized and FP16 weights; benchmarked latency/throughput vs. vLLM-style baselines to identify bottlenecks.
- Profiled CUDA kernels with **Nsight**, optimizing cache residency and stream concurrency under multi-request load.

Distributed Inference Pipeline (TP + PP) | *Python, gRPC, Grafana, CUDA*

- Implemented a multi-GPU inference pipeline combining **2-way tensor parallelism** and **2-stage pipeline parallelism** for large models.
- Developed **gRPC activation-shard RPCs** with backpressure to stabilize end-to-end throughput under bursty workloads.
- Built **Grafana + Prometheus** dashboards tracking GPU utilization, per-stage latency, NVLink/PCIe bandwidth, and queue depths.
- Optimized microbatch sizes and scheduling windows to maximize utilization across heterogeneous GPUs.

Reinforcement Learning for Robotic Control at Scale | *PyTorch, JAX, MuJoCo, CUDA, Docker*

- Trained **SAC/Transformer** grasping policies using **GPU-accelerated SDF models** for differentiable collision checking
- Leveraged **JAX vmap/pmap** for thousands of parallel MuJoCo rollouts, improving RL throughput by **3×**.
- Integrated batched SDF inference into MuJoCo simulation for high-frequency geometric reasoning during grasping.
- Containerized multi-GPU RL workflows with **Ray + Docker**, enabling scalable, reproducible experiment sweeps.