Sirish Gambhira

README.md | sirishgam001@gmail.com | linkedin.com/in/sirish-gambhira

Education

Georgia Institute of Technology — Exp: Dec 2025
*M.S in Computer Science (Machine Learning), GPA - **4.0 / 4.0*** — *Atlanta, GA*

Indian Institute of Technology, Kharagpur — Aug 2017 - May 2022
*B.Tech + M.Tech, Electronics & Electrical Communication Engineering, Minor in CS - **9.33 / 10.0*** — *Kharagpur, India*

Experience

AI Frameworks Intern | AMD Quantization — May 2025 - Aug 2025
- Proposed and implemented multi-gpu based solutions for GPTQ quantization of MoE models and model evaluation
- Bypassed GIL in python for quantization of experts and distributed the computation among GPU devices using model parallelism
- Achieved **1.65x** and **10x** speedup in quantization of Qwen1.5-MoE-A2.7B & DeepSeek-R1 with **0.5%** improvement in perplexity
- Implemented data parallelism, partitioning the evaluation dataset across ranks and achieved **5.01x** speedup in evaluation time

Research Fellow | Microsoft Research India — July 2022 - July 2024
- Primary contributor to HyWay - an interactive communication platform for in-person and remote participants.
- Built an image editor powered by CosmosDB to create/edit event-spaces. Contributed to core-fixes such as memory-leakage: in-place mutation of game-objects, re-renders: migration from react state to redux store.
- Published our work at UbiComp'23 and filed a U.S patent (Application No: 18/358,485 (2024))

Indoor User Localization and Representation
- Developed a computer-vision system to detect, track, localize and represent using multiple cameras.
- Proposed an inter-camera matching algorithm to associate identical people across different camera viewpoints.
- Optimized an open-source face detection lib (RetinaFace) and significantly reduced the end-to-end processing time of our system.
- Real world evaluations showed our mean inter-camera matching algorithm accuracy is around **90%**, mean indoor localization error is **1.037m** and the overall pipeline achieved **15 FPS**, supporting real-time operations.

Continual Learning for User State Modeling
- Developed a heuristic + ResNet based continual learning framework for determining whether a user is distracted or not using multiple data sources in a privacy-preserving manner.
- Implemented a data-processing module to collect and synchronize data from multiple sensors (e..g, cam, mic, keyboard, screen).
- Implemented an acoustic echo cancellation module to facilitate usability of the tool in a headset-free setting.
- Proposed a continual learning framework to generate annotated video data on fly and trained a ResNet to model each user.

Data and Applied Scientist Intern | Microsoft India — May 2021 - July 2021
- AI graph is a user-centric graph with user's emails, meetings, documents as nodes and topics, associated people as edges.
- Developed a graph parsing algorithm to extract user's topics and used heuristic scores to quantify user-topic relevance.
- Used Holt-Winters models for time-series analysis and visualized interactive spider & radar charts using D3.js

Selected Projects

Project Vajra — August 2025 - Present
- Implementing support to serve FP8 quantized models using our in-house inference engine Vajra
- Designed a custom weight loader to read fused fp8 quantized weights, dequantized into float16 and requantized to torch.float8 with unified scale using custom torch kernels

Stitch-LLM — Jan 2025 - April 2025
- Disaggregated prefill and decode stages of LLM inference using two different LLMs (e.g., Llama-3-3B and Llama-3-1B)
- Created a hybrid network using 'stitch-layers' to transfer KV cache between prefill and decode during inference
- Two stage training: a) stitch-layer warmup between the predicted and actual KV caches b) end-to-end LoRA based finetuning.
- Observed interpolation in **rogue** scores for different LLM series (e.g., Llama, Qwen, Minitron)

SEAM: Stitchable Efficient Architectures for ResNet Models — Jan 2025 - April 2025
- Created intermediate architectures using stitching and interpolated the compute, performance of ResNet34 and ResNet50
- **92.16%** of our stitches achieved interpolation upon fine-tuning the end-to-end network on ImageNet
- Served lighter competitive models compared to baselines **100%** of the time for a given accuracy constraint

Caching policies using L2 persistence for DNN inference — Aug 2024 - Dec 2024
- Started with hypothesis that default hardware caching policy is sub-optimal for weight-shared DNNs inference (metric: latency)
- Implemented dynamic prefetching, fetch next layer's weights during current layer's computation [CUDA stream concurrency]
- Observed latency improvement for hand-written CUDA kernels for linear and convolutional networks on RTX 3070 GPU
- Extended policies for custom PyTorch models (e.g., OFA MobileNet) and observed selective gains **8.27%** compared to default hardware policy. Justified latency performance using DRAM read/write measurements from Nsight compute.

Can DB Queries Exploit Tensor Cores? — Oct 2024 - Nov 2024
- Implemented hash join operation between tables of size 1M rows using matrix multiplication in Triton on H100 GPU
- Implemented hash lookup as an alternative approach in CUDA and compared its performance against Triton
- Results indicated that average run-time for CUDA kernel is 90.65us and for triton is 13.78ms (**150x faster**)
- Speedup achieved is due to higher occupancy of CUDA (**74.14%**) compared to Triton (**18.67%**), obtained using Nsight Compute

Parallel bitonic sort using CUDA — Aug 2024 - Dec 2024
- Implemented parallel bitonic sort algorithm over 10M array elements on L40S GPU
- Reduced memory access time by utilizing shared memory instead of global memory, improving throughput from **67%** to **93%**
- Explored host-device data transfer optimizations such as pinned-memory to reduce transfer time from **15ms** to **5ms**
- Proposed approach achieved **110x** speedup over CPU based sorting.

3D Object Reconstruction using multi-view 2D images — Aug 2021 - May 2022
- Implemented an encoder-decoder architecture to generate independent 3D point clouds from each multi-view image and fused them into a unified 3D point cloud object.
- Extracted 2D key points from the generated 3D objects for supervision, eliminating the need for labelled 3D ground truth.
- Obtained a Chamfer distance of **5.12**, comparable to the state-of-the-art single-view reconstruction result of **3.48**.

Technical Skills

Languages: Python, C++, JavaScript **Frameworks:** PyTorch, TensorFlow, CUDA, Triton, NSight Compute