Sayon Ramanan

647-390-4397 | sayon.ramanan@uwaterloo.ca | LinkedIn | GitHub | X

Passionate about building high performance systems at the intersection of hardware and software.

Education

University of Waterloo Waterloo, ON

Bachelor of Applied Science in Computer Engineering

Skills

Languages: Python, C/C++, Bash
ML Frameworks: PyTorch, TensorFlow, NumPy
Inference/Systems: MLIR, INT4/INT8 quantization, KV-cache optimization, SIMD
Tools: Git, Docker, Linux, CMake, GitHub Actions, Jenkins

Experience

Software Engineering Intern Sep 2026 – Dec 2026

onsemi *Waterloo, ON*

- Built an MCP server exposing flash, register/memory inspection and UART checks as tools, allowing agents to run the firmware debug loop autonomously across, integrated in **10+** SoC lines, cutting failure-diagnosis time by 30%
- Cut **PyTorch** inference latency up to **5.6x** by splitting layers across 8 on chip **neural-network-accelerators** (**Python**, **C**, **MLIR**), bit-exact on silicon
- Worked across **quantization** flow of a **PyTorch**-to-silicon compiler (PTQ, QAT, ONNX Export) creating a validation pipeline against a FP32 golden model
- Designed **Jenkins** pipelines and containerized build environments (**Docker**) for an embedded SDK, replacing manual HAL checks with regression tests on compiled firmware.

Machine Learning Engineer Intern Jan 2026 – Apr 2026

Portable Intelligence Inc. *Toronto, ON*

- Trained deep **reinforcement learning** policy networks (**PyTorch, stable-baselines3**) to solve warehouse slot assignment as a sequential part-placement problem, reducing customer order batch completion time by **35%**
- Re-architected the policy network from an MLP to a **transformer** for sequential slot selection, leveraging self-attention to improve travel time by **15%**
- Built an evaluation platform to benchmark trained RL policy networks, running rollouts across vectorized environments via **Tensorboard**

Research Engineer Jan 2026 – August 2026

Wat.AI *Waterloo, ON*

- Deployed a **transformer** policy (ACT) on an NVIDIA Jetson for closed loop control of a 6-DOF SO-101, at **30Hz**, running inference from a dual camera input.
- Published training policies and demonstration (**60+** episodes) to Hugging Face Hub- allowing reproducibility on any device with zero local setup

Projects

On-Device Speculative Decoding on Raspberry Pi 5 | *C++, Python, HuggingFace* | 2026

- Wrote performance-tuned INT8 matmul **kernels** in **C++** using **ARM** NEON intrinsics, benchmarked against llama.cpp on **ARM** Cortex-A76
- Implemented speculative decoding on a single Raspberry Pi 5, running two LLMs across separate CPU cores
- Applied grouped **quantization** on HuggingFace model weights, reducing memory footprint by **75%**

Flash Attention (from Scratch) | *Python, PyTorch, NumPy* | 2026

- Implemented a tiled attention forward pass that streams KV blocks with online softmax, avoiding $O(N^2)$ score-matrix materialization while exactly matching standard attention
- Added a backward pass via **PyTorch** custom autograd, recomputing attention scores in tiles to keep training memory linear in sequence length