Bhavik Upadhyay

Email: bkupadhy@usc.edu
Mobile: (650) 441-8815
Github: github.com/bhavikupadhyay
LinkedIn: linkedin.com/in/bhavik-upadhyay

Education

University of Southern California — Los Angeles, CA
Master of Science in Computer Science — Aug. 2023 – May. 2025
- CGPA: 3.91 / 4.00
- Relevant Coursework: Machine Learning, Computer Vision, NLP, Robotics, Algorithms, Operating Systems, Web Development

CVR College of Engineering — Hyderabad, India
Bachelor of Technology in Computer Science and Information Technology — Aug. 2019 – May. 2023
- CGPA: 3.74 / 4.00 | Department Gold Medalist
- Relevant Coursework: Data Structures, Graph Theory, OOP, Distributed Systems, Computer Architecture

Experience

Backend and Systems Engineer — Sep. 2025 – Present
Easley Dunn Productions, Inc.
- Built a C# transaction subsystem in Unity governing card pack purchases and inventory state, with atomic flows and rollback logic to keep the in-game economy consistent across concurrent sessions
- Cut screen transition latency by **13%** by consolidating 4 fragmented Unity UI screens into a single canvas with decoupled event handling and centralized state
- Automated CI/CD pipelines for a multi-branch Unity monorepo on GitHub Actions

HPC Researcher — Sep. 2024 – May. 2025
GRIDS Lab @ University of Southern California
- Shipped **OTTER**, a C++17 reverse-mode autodiff library with dynamic compute-graph construction and **16+ differentiable ops** spanning matmul, batched broadcast, reductions, views, and elementwise primitives
- Wrote both memory allocators from scratch: an mmap pool with posix_memalign for host memory, and a pooled cudaMalloc allocator with reactive OOM eviction and stream-async kernel launches for device memory
- Implemented the autograd engine on a 4-worker thread pool for parallel topological traversal, with mutex-protected gradient accumulation into shared leaf tensors via passkey-gated buffer access
- Decoupled hardware paths via templated dispatchers, a functor-based CUDA kernel registry, a stable Backend interface, and CMake/Ninja conditional compilation; new backends drop in with zero changes to Tensor or autograd code

Teaching Assistant — Oct. 2023 – Dec. 2024
University of Southern California
- Mentored **850+ graduate students** across Database Systems, Applied NLP, and Machine Learning & Computational Physics, and standardized grading rubrics with faculty for a **7% improvement** in average student scores

Projects

CoverageAgent: 7-Agent Test Generation Pipeline — Apr. 2026
- Designed a 7-agent LangGraph system that targets uncovered branches in public Python repos, generates pytest tests, sandbox-verifies them in E2B Firecracker microVMs, and commits; lifted python-slugify branch coverage by **33.33%**
- Replaced RAG with Jedi-based static analysis for context retrieval and made the test-quality evaluator fully deterministic (`ruff F821` + import plausibility) after v3 benchmarks showed LLM grading approved tests that crashed at runtime

High-Performance Triton GPU Kernels — Jan. 2026
- Authored **22 custom OpenAI Triton kernels** across 7 progressive phases covering elementwise ops, reductions, tiled matmul, FFT, convolution, and Flash Attention v2, benchmarked against PyTorch/cuBLAS with roofline analysis
- Hit **cuBLAS parity within 1%** on tiled matmul at N=4096, a **1.69× fused-kernel speedup** matching the theoretical memory-traffic ratio, and a **∼50× speedup** on parallel-scan cummax over `torch.cummax`

Weenix: Unix-Like x86 Kernel — May. 2024
- Constructed core kernel primitives (process scheduling, thread synchronization, signal handling, context switching) plus VFS and Virtual Memory subsystems enforcing kernel-level guarantees for `fork`, `mmap`, and `open`
- Led a 4-person team through a critical race condition discovered hours before release by devising a split-debugging strategy across scheduling and VFS

CIFAR DDPM Image Generator — Feb. 2025
- Trained a denoising diffusion model from scratch in PyTorch on CIFAR-10 with a custom UNet (sinusoidal time embeddings, residual attention blocks, configurable noise scheduler), reaching **FID 87.94** on class-conditional generation
- Deployed to Hugging Face Spaces after int8 post-training quantization and ONNX Runtime export, cutting framework dependencies and container footprint

Technical Proficiencies

Languages: Python, C++, C, C#, TypeScript, JavaScript
ML & AI: PyTorch, TensorFlow, CUDA, Triton, HuggingFace, LangChain, LangGraph, ONNX, OpenCV, ChromaDB
Web & Infra: FastAPI, Node.js, Express, Django, React, Next.js, Angular, Docker, AWS, MongoDB, CMake, GitHub Actions
Key Skills: AI Agents, LLMs, RAG, GPU Programming, Kernel Optimization, Systems Programming, Multithreading, Distributed Systems, Deep Learning, Generative AI, NLP, Computer Vision, Quantization, MLOps, Full-stack Development, Microservices