Keegan Smith

+1 (832) 773-8754
keeganasmith2003@gmail.com
keegancodes.com | github.com/keeganasmith

Education

Texas A&M University — College Station, TX

Master of Science — Computer Science / GPA: 3.9 Jan 2025 — May 2026

Bachelor of Science — Computer Science / GPA: 4.0 Aug 2021 — Dec 2024

Selected Coursework:

Quantum Algorithms	Deep Learning (PyTorch, Tensorflow, Pandas)	Software Engineering (Ruby on Rails)
Computer Architecture (C++)	Parallel Algorithms (C++, Open MPI, YGM)	Computer Networking (Visual C++)
Operating Systems (C++)	Machine Learning (PyTorch)	Programming Languages (Java, Scheme)

Work Experience

- **High Performance Computing (HPC) Software Engineer** *May 2023 – Present*
  Texas A&M University High Performance Research Computing College Station, TX
  - Benchmarked multi-core FFT and sparse matrix multiplication (**FFTW**, **oneMKL**) in C++ on Intel Optane configured as swap; analyzed performance tradeoffs for memory-bound HPC workloads
  - Migrated Fermilab FPGA software simulation workflows from Kubernetes to Slurm, improving reproducibility and reducing researcher setup overhead
  - Built an Open OnDemand interactive app for distributed local LLM inference across multi-node jobs using Python (4 Intel GPU Max 1100s per node)
  - Shipped an Open OnDemand app to automate ML pipelines with **AutoGluon** on Slurm; standardized job specifications and reduced experiment setup time
  - Automated ticketing workflows by integrating Slack with the Slack API; built a CLI workflow engine enabling ticket state changes without code changes
  - Resolved HPC support tickets across software troubleshooting, job scheduling, and environment configuration for diverse scientific workloads
  - Developed on RHEL 8 systems across shared HPC infrastructure
- **iOS Developer** *May 2023 – Present*
  AKW Ventures College Station, TX
  - Built and maintained backend systems for "College Football: Rapid Roster" iOS app (Swift, JavaScript, CloudKit JS)
  - Scaled the app to thousands of active users and $2,500 monthly recurring revenue
  - Developed a full-stack sports betting analytics platform that detects arbitrage, +EV, and middles opportunities
  - Designed cloud deployment on AWS (Lambda, SQS, RDS/Postgres, ECS/Fargate, ECR) with Docker + GitHub Actions CI/CD
  - Implemented a C++ datafeed pipeline, a Go REST API backend, and a SwiftUI frontend
  - Built "TEC Call Auditor" using AI agents to grade phone calls; achieved 96% production accuracy via prompt engineering and agent orchestration
- **Software Engineer Intern** *May 2024 – Aug 2024*
  The Coca-Cola Company Atlanta, GA
  - Built a Vue.js + Flask web app for segmenting and analyzing 8,861 retail outlets using Python and Pandas
  - Developed a promotion recommendation algorithm generating 124,054 outlet-relevant promotions to confirm targeting and reduce manual analysis

Research

- **Parallel Bridge Finding** *January 2025 - Present*
  Texas A&M University, under Dr. Roger Pearce
  - Developed and iterated upon a novel parallel bridge finding algorithm (graph theory)
  - Achieved near linear scaling up to 768 cores - 15.36 times as many cores as the current literature supports
  - Distributed computing on Lawrence Livermore National Labs systems, notably Dane
  - Implemented in C++ using the YGM library (built on OpenMPI)

Projects

- **Temporal Frequency Analysis on CPU, GPU, FPGA** *August 2024 - December 2024*
  Texas A&M University capstone project done in collaboration with Sandia National Labs
  - Optimized an existing temporal frequency analysis algorithm for CPU, GPU, and FPGA architectures
  - Won 3rd place in the Texas A&M Engineering Project Showcase Fall 2024
  - Our GPU implementation achieved the highest throughput, achieving a 125x throughput improvement over the original FPGA implementation
  - Increased throughput of FPGA implementation by 4.23x
  - Used C++, Vitis HLS, CUDA, OpenMP, JavaScript, Express.js, React.js
- **Push Battle AI** *February 2024*
  Texas A&M Datathon

- Developed a minimax implementation for the push battle board game, and used distributed computing (MPI) to create a massive database of opening moves
- Won 2nd place out of 600 competitors

- **M-Height Approximation** *May 2025*
  Texas A&M University CSCE 636 (Deep Learning)
  - Created a large training dataset (approximately 100 million m height problems and solutions) using OpenMPI
  - Trained a dense neural net with 1 million parameters on multiple A30 GPUs
  - Achieved a log loss of 1.20
  - Used PyTorch and OpenMPI
- **Twitter Huffman Compression Bot** *April 2022*
  Personal Project (Python, Tweepy, Twitter API, Heroku)
  - Built a Twitter bot that compresses and decodes user messages using Huffman coding via mention-triggered commands
  - Integrated the Twitter API with Tweepy to monitor mentions, parse keywords (compress/decode/time), and publish automated replies
  - Implemented Huffman encoding/decoding with a custom tree construction pipeline (list-based priority structure) and prefix-code decoding
  - Added id-based deduplication to prevent duplicate replies and ensure consistent bot behavior across polling cycles
  - Deployed the service on Heroku for continuous execution and remote operation

Technical Skills

Languages: C++, Python, Go, Java, Bash, Swift (SwiftUI), JavaScript, Scheme
HPC / Systems: MPI (OpenMPI), OpenMP, CUDA, Slurm, YGM, FFTW, oneMKL, Vitis HLS, Linux (RHEL 8), Unix
Machine Learning / AI: PyTorch, TensorFlow, Pandas, AutoGluon, LLM Inference, Prompt Engineering, Agent Orchestration
Web / Backend: Flask, FastAPI, Vue.js, React.js, Express.js, REST APIs, CloudKit JS, Ruby on Rails
Cloud / DevOps: AWS (Lambda, SQS, RDS/Postgres, ECS, Fargate, ECR), Docker, Singularity, GitHub Actions, CI/CD, Heroku
Databases: PostgreSQL, DynamoDB, MongoDB
Tools: Git, GitHub, Vim, GoogleTest, Tweepy, Twitter API
Methodologies: Object-Oriented Programming (OOP), Functional Programming, Test-Driven Development (TDD), Agile