

Eric Tang

erictang000@berkeley.edu | 408-410-3070 |  /erictang000 |  /erictang000 | erictang000.github.io

EDUCATION

UNIVERSITY OF CALIFORNIA, BERKELEY

BS IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

AUGUST 2018 - MAY 2022

GPA: 3.82 / 4.0, Major GPA: 3.92 / 4.0

Selected Coursework: Applications of Parallel Computers, Deep Neural Networks, Databases, Machine Learning, Full Stack DL, Optimization Models, Computer Architecture, Efficient Algorithms, Computer Security, Operating Systems, Data Structures
Planning to remain at Berkeley for 5th year MS program with expected graduation in Spring 2023.

SKILLS

Languages: Java, Python, C/C++, Golang, SQL, CSS, HTML, Swift, Javascript

Tools: Git, Linux, XCode, Numpy/Pandas, OpenMP, MPI, Django, CUDA, MongoDB, Flask, Docker, Apache Nifi, AWS

Machine Learning: Pytorch, Keras, OpenCV, Sklearn, CNNs, Transformers, RNNs/LSTMs

EXPERIENCE

ACCENTURE LABS

JUNE 2020 - AUGUST 2020, JUNE 2021 - AUGUST 2021

Technology R&D Intern

- **Summer 2021:** Built backend and data pipelines for a knowledge graph based digital twin platform using **TimeScaleDB**, **Apache Nifi**, and **Stardog**, with probabilistic machine learning in **PyMC3**.
- Created a machine learning model serving API for training and inference on time series data using **Docker** and **BentoML**, modelling in **tsfresh**, **sklearn**, and **PyTorch**.
- **Summer 2020:** Developed software architecture in **Python** for a generative design project with a warehouse layout use case, establishing parameterization based on model and expert specs, and writing a layout generation algorithm with intended application to future company projects. Inventor on pending patent.
- Worked on Bayesian black box optimization for layout generation using **sklearn**, and built an evaluation function of layouts based on congestion modelling along the generated graph using **networkx**.

BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

AUGUST 2020 - PRESENT

Undergraduate Researcher

- Worked on using computer vision action recognition models to predict emotional response and human preferences for ranking video data, and using transformer based models to solve difficult math problems with Dan Hendrycks via URAP with Prof. Dawn Song.
- Lead data labeling and filtration using **AWS MTurk** and **Javascript** on 60,000 scraped videos.
- Trained video understanding models for adversarial filtration and model assessment using **PyTorch**, managing nodes for machine learning model training on GPU cluster with Slurm.
- Fine tuned and evaluated large language models (GPT-3, GPT-2, AI21 Jurassic-1) on various downstream tasks.

UNIVERSITY OF CALIFORNIA, BERKELEY

JANUARY 2020 - PRESENT

Lead Software TA for CS 61B - Data Structures

- Worked as lead software infrastructure TA to help develop course autograder software using **Docker**, **Java**, and **Make**, and course grading tool **Beacon**, using **SQLAlchemy** and **Flask** and deployed using **Dokku**, serving 1000 students per semester.
- Manage small team of newer TAs in use and development of course infrastructure and software engineering workflows.

UC BERKELEY PHYSICS - CROMMIE GROUP

APRIL 2019 - PRESENT

Undergraduate Researcher

- Designed and trained convolutional neural networks in **Keras** for classification of subdiffusive particle behavior on sparse experimental data, achieving SoTA accuracy of 85% for sparse time series diffusivity classification.
- Modified models for generating novel visualizations for particle tracks using class activations without loss of accuracy, and demonstrated alignment with expected physical diffusive phenomena.

PROJECTS

GPU-BSW WORK STEALING

MAY 2021

C++, CUDA, OpenMP, HPC

- Implemented work stealing for a GPU implementation of the Smith-Waterman sequence alignment Algorithm using **CUDA** and **OpenMP** in **C++** for heterogenous computing environments. Final project for CS 267 - Parallel Computing (grad course).
- Integrated SIMD CPU version of Smith-Waterman with batched GPU/CUDA based Smith-Waterman to work in parallel off of a shared global queue. Experimented with different work sharing/stealing approaches, and benchmarked performance on high performance computing clusters.

TINY IMAGENET CLASSIFIER

APRIL 2021

Pytorch, Slurm, Jupyter

- Built custom architecture for classification on Tiny ImageNet based on Vision Transformers using **PyTorch**. Achieved 85% top 1 accuracy, and explored use of different attention mechanisms, data augmentation methods, and pretrained base models for improving out of distribution robustness. Computer Vision final project for CS 182 - Deep Learning.