ALEC MACLEAN GUNNY

Associate Researcher at MIT, IAIFI Junior Investigator

@ alec.gunny@gmail.com (310) 600-3175 San Francisco, CA github.com/alecgunny alecgunny.github.io

EXPERIENCE

Massachusetts Institute of Technology/LIGO

Machine Learning Research Associate — Nov 2020 – Present

- Using neural networks to solve problems in gravitational wave (GW) physics via product-oriented applications with a focus on scale, strong physical priors, and interactive analysis using Bokeh.
- Developing a robust MLOps ecosystem built around PyTorch, TensorRT, and Triton to facilitate GPU-accelerated applications in both cloud and HPC computing environments.
- Leading team of grad students and post-docs in research on ML-based binary blackhole merger detection; building robust workflows for distributed development on GitHub.
- Giving talks to summarize findings, challenges, and best practices to foster international collaboration and arrive at domain-wide best practices for the use of ML models in GW physics.

NVIDIA

Applied Machine Learning Scientist — Jan 2020 – Nov 2020

- Worked on GPU-accelerated tabular data processing library NVTabular, focusing primarily on integration with TensorFlow data loading APIs to accelerate tabular model training by a factor of 10-20x.

Solution Architect/Senior Solution Architect — Sep 2017 – Jan 2020

- Collaborated with data science and infrastructure teams from large consumer internet companies to build scalable, GPU-accelerated deep learning systems for both training and inference.

Children's Hospital Los Angeles

Data Scientist — Apr 2016 – Sep 2017

- Researched recurrent neural networks to model asynchronously and irregularly sampled EMR data from patient stays in the pediatric ICU and built model visualization applications for collaboration with physicians.

Arete Associates

Scientist — Aug 2014 – Apr 2016

- Combined signal and image processing techniques with machine and deep learning to solve a range of detection and regression problems.

PUBLICATIONS

Hardware-accelerated Inference for Real-Time Gravitational-Wave Astronomy

Nature Astronomy — May 2022

Describing an efficient computing paradigm for leveraging neural networks at scale in GW physics and benchmarking its use in multiple contexts.

End-to-end acceleration of machine learning in gravitational wave physics

Fast Machine Learning for Science Workshop — October 2022

Brief talk identifying challenges in building ML-based GW applications and introducing libraries of tools meant to address them.

A Software Ecosystem for Deploying Deep Learning in Gravitational Wave Physics

FlexScience 2022 — July 2022

Discussing the requirements of a particular real-time, ML-based GW application and outlining plans for its deployment.

SKILLS

Python
Bash
JavaScript

- Python Libraries

TensorFlow | Keras | PyTorch | NVTabular | Scikit-Learn | Pandas | CuDF | Bokeh | OpenCV | Flask | TensorRT

- Cloud Computing/Orchestration

Google Cloud | AWS | Docker | Docker Compose | Kubernetes | Singularity/Apptainer

EDUCATION

B.Sc. in Engineering Physics

University of California Berkeley

Aug 2010 – May 2014

Graduate with honors

PROJECTS

- Detecting binary blackhole mergers from time-domain GW strain with convolutional neural networks
- Using autoencoder-adjacent models to subtract environmental noise from GW strain to increase sensitivity
- Visualization and analysis of model performance from current `main` branch of noise subtraction model
- `ml4gw` Python library containing efficient PyTorch implementations of common GW operations
- Set of Python libraries called `hermes` for simplifying the export, acceleration, and deployment of trained models using Triton Inference Server
- Presentation outlining the advantages of inference-as-a-service in gravitational wave physics
- Command line utility `pinto` for managing and executing ML pipelines across complex environments leveraging a hybrid of Conda and Poetry
- `typeo` (TY-poh) utility library for turning type-annotated functions into command-line executables with robust argument support and TOML config parsing