

Matthew Hoffman

matthewhoffman.dev



SKILLS

- **Languages:** Python, C, C++, Java, TypeScript
- **Computing:** PyTorch, NumPy, multiprocessing, OpenCV
- **Data Analysis:** pandas, SQL, scikit-learn, matplotlib
- **Cloud:** AWS

EXPERIENCE

- **Protopia AI** San Francisco, CA
Research Engineer May 2022–Present
 - **Distributed PyTorch Training Library:** Rearchitected and owned the entire training library used to train all models at the company. Parallelized training from scratch, redesigned the class hierarchy, improved data modeling and configuration using pydantic, built tools to visualize vision model parameters & activations. Enabled unique training algorithms and wrote tests to ensure correctness.
 - **Core Software Development Kit:** Defined, architected, and extracted the company's core technology into a minimal SDK to be integrated into generic training loops (such as huggingface, PyTorch Lightning). Used advanced Python typing to build a simple user interface, allowing for intuitive integration with only a few extra lines of code.
 - **Real-time Video Processing Pipeline:** Architected a real-time video processing pipeline framework and deployed it for multiple customers; parallelized using Python multiprocessing and shared memory; profiled code to identify bottlenecks; read video & webcam input and formatted & displayed output using OpenCV; implemented face-tracking and five-crop algorithms in NumPy, vectorized KNN using PyTorch; created rotating file handlers for output logs & videos.
 - **Engineering Excellence:** Standardized engineering practices; configured build pipelines to enforce type hints using pyright & mypy, linted code and docstrings using pylint & ruff, enforced 100% code coverage, used conventional commit messages for semantic versioning, created PR and issue templates.
- **Amazon.com, Inc.** Austin, TX
Software Development Engineer Jun 2020–May 2022
 - **Accounting Configuration:** Delivered the accounting requirements for multiple cross-region accounting projects as a solo developer; responsible for the clarification of requirements, design, implementation, testing, verification, communication of timelines with stakeholders, and post-launch monitoring.
 - **Workflow Orchestrator:** Designed an accounting workflow orchestrator using AWS Step Functions, Lambda, SQS, SNS, & DynamoDB.
 - **VS Code Language Server Extension:** Created a Language Server extension in TypeScript to improve the developer experience when writing accounting configuration in Amazon Accounting's custom language. Implemented syntax diagnostics, redefinition error detection, keyword definition hovers, dynamic go-to definition, code-completion, and format on-save.
 - **Automated CLI Generator:** Created a Python module to automatically create command-line parsers by statically analyzing the entry point function and its documentation. This module simplified the developer experience and served as a central place to register and share commonly used scripts between team members.
 - **Web Scraping & Requests Automation:** Created a Python package to scrape and automate requests to Amazon-internal websites. Used this package to create a script to automate an event failure process saving 1 developer-day per week. This package also saved multiple days per project when used to collect and validate transaction examples during end-to-end testing.
- **Blue Cross and Blue Shield of IL, MT, NM, OK & TX** Richardson, TX
Data Science Intern Jun 2019–Sep 2019
 - **Interactive Map Visualization:** Created a Python script to generate interactive choropleth maps. Combined multiple publicly available data sources to visualize insurance coverage by zip code and identify candidate locations for BCBS' Care Van immunization program.
 - **Case Notes Text Mining:** Used regular expressions and NLP models to analyze case management notes to identify member outcomes for use in program evaluations.

OPEN SOURCE

- **einops-style rearrange in funtorch:** Implemented einops.rearrange natively in funtorch. Implementation uses the parsed string to dynamically build functions to perform the rearrange operation using PyTorch first-class dimensions.
- **huggingface transformers gradient synchronization bug:** Discovered and fixed a bug where torch DistributedDataParallel was silently not synchronizing gradients for transformers models when using static graph.

EDUCATION

- **University of Texas at Austin** Austin, TX
B.S. Computer Science; GPA: 3.75/4.00 Aug 2015–May 2020