Echo Nolan
Machine Learning Engineer

Links online at echonolan.net/resume/cv.html or via QR code

✉ echo@echonolan.net
⌁ http://echonolan.net
New York, NY USA

Professional Experience

December 2024 - March 2025 :
AI Safety Research Engineer (3 month contract). *Redwood Research*. Contributed to white box monitoring for AI control through several technical improvements. Engineered and optimized data generation pipelines for synthetic unsafe-behavior datasets, identifying and resolving performance bottlenecks that enabled more efficient generation of our ~11k examples. Refined prompt engineering techniques to significantly improve filtering accuracy, and enhanced model training processes for both detection probes and fine tuned language models that simulate problematic behaviors. The dataset improvements I delivered supported the team's research on probe generalization and robustness to adversarial examples as well as the research of external teams. My work on the finetuned models led to successfully implementing conditional unsafe behavior and evading detection probes in those models. Python, PyTorch, Hugging Face Transformers and Accelerate, and DeepSpeed.

January 2020 - December 2024 :
Sabbatical. Following the successful launch of the cryptocurrency I worked on at O(1) - Mina.

December 2018 - January 2020 :
Protocol Engineer. *O(1) Labs*. Helped build a new cryptocurrency. Worked primarily on security issues, performance and peer to peer networking. Achievements include: designed and implemented the trust system for DDoS mitigation. Rebuilt the transaction pool for efficiency and resiliency. Made the ledger Merkle synchronization process more efficient and secure. OCaml.

July 2017 - November 2018 :
Software Engineer. *Signal Vine Inc.*. Worked on the backend of a two-way text messaging platform in the education technology space. Achievements include: Added support for new upstream providers, allowing the company to serve clients in foreign countries and saving the company ~33% on messaging and phone number costs. Improved SQL query performance by 40x in some cases, making users happy and enabling further growth. Haskell, Scala, PostgreSQL and MySQL.

Open Source Projects

May 2023 - Ongoing :
⌁ **txt2img-unsupervised**. Personal research project building a text-to-image model without directly using any labeled image data. Relies entirely on CLIP for the association between text and images. Results so far are promising. A blog post about the project is ⌁ here. Developed the initial concept, gathered a 35M image dataset (⌁ blog post), ported a PyTorch VQGAN implementation to JAX, built an efficient preprocessing pipeline, trained baseline models, designed and implemented experimental models, optimized hyperparameters, designed and implemented a system for efficient sampling from the subset of a discrete set of unit vectors that are contained within a spherical cap. Built a little gallery SPA to showcase the dataset. Model is a transformer based deep neural net. Code mostly in Python, using JAX and Flax.

January 2024 - February 2024 :
⌁ **Infinidata**. A Python library for working with arbitrarily large datasets, created to support the spherical cap sampling system for txt2img-unsupervised. Written in Rust. Supports shuffling, arbitrary permutations, slices, concatenation, and more.

October 2016 - June 2017 :
Beescheduler. A single page web application to schedule goals using the goal tracking service Beeminder's API. Personal project. Frontend using React, backend using Node.js and the Serverless Framework. Unofficially sunsetted as of 2024.

Various :
A webapp allowing users to rank software, written to ⌁ empirically research the effects of programming language choices. An ⌁ implementation of the ADEL algorithm for finding minimal subsets of a given input satisfying an arbitrary upwards-closed property. CLI script to forward ports using the Private Internet Access VPN service. A mostly-complete BitTorrent client, ⌁ Whiteout.

Open Source Contributions

June 2014 - October 2016 :
⌁ **Idris**. A programming language with dependent types, implemented in Haskell. ⌁ Many contributions including improvements to the dynamic linker, parser, build system, terminal support, continuous integration and documentation. Wrote bug reports and reduced others, helped people on mailing lists and Stack Overflow.

March 2016 - June 2016 :
⌁ **GHC Libraries**. The built-in libraries for the flagship Haskell compiler. Fixed an especially hairy bug in the Windows networking functionality. The build tool Stack would hang intermittently when downloading dependencies. Traced it down to http-conduit, then to base. Submitted a ⌁ patch to GHC which was merged May 19 2016. Submitted a workaround patch to the network library. Wrote a ⌁ workaround patch to the network library. Wrote a ⌁ blog post describing the process.

Various :
⌁ Support for using the Web3 Ethereum API in the browser, in Rust. Small changes to Cabal, Stack, the Selenium Docker images, Haskell's network library, the NixOS packages, serverless-offline and serverless-simulate. Other miscellaneous work.