# Aidan Ewart

baidicoot.github.io | GitHub

Email: aidanprattewart@gmail.com

## Summary

I am a 2nd year undergraduate student studying Mathematics at the University of Bristol. I am an experienced programmer (8+ years) and am proficient in a number of languages. I do independent machine learning safety research, and am a co-first author on a paper accepted into ICLR 2024. My main academic interests are in language model interpretability, ensuring the safety of frontier machine learning models, and programming language theory.

## Technical Skills

| | | |
|---|---|---|
| **Languages** | : | Python, Haskell, Rust, C, C++, JavaScript, x86 Assembly, Lua |
| **Frameworks** | : | PyTorch, HuggingFace Accelerate |
| **Other Software** | : | Git, Copilot, Windows, Linux, LaTeX, SSH |

## Education

**University of Bristol** — Bristol
*MSci Mathematics* — *81.3% first-year average*

**Royal Grammar School** — Newcastle upon Tyne
*A-levels in Maths, Futher Maths, Physics, Computer Science* — *A*A*A*A**

**Royal Grammar School** — Newcastle upon Tyne
*GCSEs including triple sciences, Maths, Further Maths* — *99999999887*

## Publications

**Sparse Autoencoders Find Highly Interpretable Features in Language Models** — View on ArXiv
*Conference Paper, ICLR 2024*
*ATTRIB Workshop Paper, NeurIPS 2023*

- Co-first author
- Involved coordinating with Anthropic and OpenAI teams working on similar research directions
- Cited by the recent Anthropic paper 'Towards Monosemanticity: Decomposing Language Models With Dictionary Learning'

## Select Projects

**Functional Programming Language Compiler** — Source Code
*Haskell, x86 Assembly, C*

- Implemented a compiler for a Lisp-like high-level programming language
- Frontend includes Hindley-Milner typechecking and inference, a module/imports system, compilation with continuations
- Backend includes program optimisation, register allocation, compilation to x86 assembly and C

**Proof Assistant** — Source Code
*Lua, Haskell*

- Implemented a theorem-proving DSL for Lua
- Proof assistant uses a Martin-Löf style type system complete with type inference via unification
- Includes a customisable notation system in the style of Coq

## Other

- Attended the NeurIPS 2023 and Principles of Programming Languages 2021 conferences
- Co-run the Bristol AI Safety Center, a small student research group for AI safety in Bristol
- Treasurer for EA Bristol student group