Justin Meimar

meimar@ualberta.ca GitHub

Systems Software Engineer | Edmonton, Alberta, Canada

University of Alberta - MSc. Computer Science *Sep 2025 – Aug 2027*
Researching JIT compilers and techniques in collaboration with Mozilla. Designing mechanisms for reusing deterministic and statistically JIT code in an ahead-of-time setting.

University of Alberta - BSc. Spec. Computer Science *Sep 2021 – April 2025*
GPA: 3.7 / 4.0
Key Classes: Algorithms, Compilers, Software Engineering, Operating Systems, Computer Vision & Machine Learning.

EXPERIENCE

Tensor Compiler Intern | Huawei *May – Aug 2025*

Edmonton, Alberta

- Designed scheduling algorithms using hardware heuristic-guided search to efficiently lower tiled computation graphs to custom matrix accelerators.

Teaching Assistant | University of Alberta *Jan 2023 – Dec 2025*

Edmonton, Alberta

- CMPUT 415 Compiler Design (Head TA) - F24, F25: Teaching students about LLVM, MLIR and other compiler skills.
- CMPUT 429 Computer Architecture II - W24: Helped students learn and write RISCV assembly code.
- CMPUT 229 Computer Architecture I - W23

Research Assistant | Compiler Design Optimization Lab *May – Aug 2024*

Edmonton, Alberta

- Researched modern compiler techniques such as copy patch for JIT compilation.
- Debugged and upgraded solution compilers to LLVM 18, introduced new dialects and lowerings.

Full Stack Intern | Bits in Glass *May – Aug 2023*

Edmonton, Alberta

- Developed an internal web tool in Python and managed deployment on Heroku with Docker.

PROJECTS

Hack GPT Edmonton | 1st Place *July 2023*

- Hackathon - Worked in a team of four to build an interactive information retrieval system for Real Estate listings using Vector-Embeddings and LLM (large language model) APIs.

Home Server | System Administration *June 2023*

- Managed a small server with NGINX to host services over SSH and HTTPs.

Gazprea Compiler | 1st Place *Fall 2022*

- Worked in a team of four to create a tournament winning compiler; focused on type-checking, type-promotion, and control flow implementation using LLVM IR-builder.

STACK

Skills / Fluent in UNIX - Strong written and verbal communication - Persistent problem solver

Tools / Git - CMake - Docker - Nginx - ANTLR - vim - ssh - gdb - valgrind

Languages / C/C++ - Python - JavaScript & TypeScript - Bash - awk - Rust - SQL