

roife

📞 [REDACTED] | ✉ roifewu@gmail.com | 🌐 roife | 🌐 roife.github.io | 📍 Hong Kong

Education

Nanjing University

2023.09 - 2026.06 (expected)

Master's Degree in *Computer Science and Technology* | Pascal Lab. Tutor: [REDACTED] Li | Focus on PL and Program Analysis

Beihang University

2019.09 - 2023.06

Bachelor's Degree in *Computer Science and Technology* | GPA 3.84/4.00

Work Experience

NVIDIA OCG

2025.02 - Present

GPU Compiler LLVM Backend Intern

- Responsible for unifying the vectorizer between NVIDIA GPU graphics compiler and NVVM, ensuring the graphics compiler's vectorizer aligns with LLVM upstream
 - Added support for graphics intrinsics in the new vectorizer while minimizing divergence from upstream;
 - Ported several memory access vectorization optimizations to the new vectorizer, including offset gap filling, etc.;

Rust Foundation Fellowship Program

2024.09 - 2025.09

Rust Foundation Open Source Community Grant

As one of the rust-analyzer maintainers, responsible for maintaining rust-analyzer (the official Rust IDE).

- Ranked **21/972** in contributors; resolved **70 issues**; participated in issues handling, meeting discussions, PR reviews, and other maintenance work across most project modules;
 - Implemented control flow navigation, snapshot test updates, and other features, while participating in bug fixes;
 - Wrote a **SIMD** implementation for the line breaking module for ARM NEON, achieving a **6.5x** speedup on ARM platforms;
 - **Emergency incident response for v0.3.1992**: 4 hours after release, the community discovered a critical bug causing resource exhaustion and process blocking. I identified the algorithmic issue in **3 hours** and designed a new algorithm as fix. This emergency fix controlled the incident's impact, preventing disruptions for global Rust developers.
-

Awards

- 2022 **National Scholarship** (ranked 1/195 in the major), **Outstanding Graduate** of Beihang University
 - **First Prize** in the NSCSCC Compilation System Design Competition (Huawei Bisheng Cup) 2021, ranking 2nd overall.
 - **First Prize** in the Lanqiao Cup C++ Programming Contest (Beijing Division) and **Third Prize** in the National Finals
 - Additionally awarded over ten provincial and university-level awards and scholarships
-

Projects

Vizsla

🌐 roife/vizsla (WIP)

Modern IDE for Chip Frontend Design · Master's Thesis Project

Rust / SystemVerilog

- Based on **incremental computation** architecture, implemented a semantic analysis framework and IDE infrastructure for SystemVerilog, aiming to provide modern IDE features for chip design;
- Project achieves **industry-leading standards** in functionality, performance, and usability: implemented **dozens of** modern IDE features for SystemVerilog including code navigation, semantic refactoring, code completion, semantic highlighting, code diagnostics, etc., with **millisecond-level** latency through incremental semantic analysis;
- Based on the Language Server Protocol, compatible with VS Code, Emacs, NeoVim and other mainstream editors.

Ailurus


🌐 roife/ailurus (WIP)

Experimental Programming Language and Toolchain Design

Rust

- Based on **Martin-Löf type theory**; supports **bidirectional type checking**, **dependent types**, pattern matching, indexed inductive types, module system, and other features;
- Uses Normalization by Evaluation for equivalence checking, implements propositional equality;

Ayame

 [No-SF-Work/ayame](#)

Compiler from SysY (C subset) to ARMv7 · Bisheng Cup Competition Project (Collaborative)

Java / LLVM-IR / ARM

- Primarily responsible for backend optimizations targeting Machine IR, including graph-coloring based **iterative register merging**, **instruction scheduling**, and peephole optimizations;
- Also handled project testing and DevOps, setting up testing workflows with Docker and GitLab CI, and writing Python scripts to automatically analyze test results;
- Project ranked **2nd overall** in the competition, achieving **1st place in nearly half of the testcases** and outperforming gcc -O3 and clang -O3 on 1/3 of the examples.

LLVM-Lite










 [roife/llvm-lite](#)

Lightweight Edge-side Compiler for Neural Network Operators · Undergraduate Thesis Project

C++ / LLVM-IR

- Project aims to utilize known **shape information** from edge inference devices to perform **secondary optimization** on deep learning operators, reducing runtime spatial and temporal overhead;
- Project includes a lightweight compiler and a trimmed LLVM codegen module; successfully reduced inference time by 6% and binary file size by 38% for convolution and softmax operators;
- Implemented **parse-time optimizations** that reduced compilation time by 60% and memory usage by 60%; received **excellent** evaluation for the thesis.

🔗 Open Source Contributions

- **Rust Organization** (rust-analyzer contributors team) member, primarily maintaining  [rust-lang/rust-analyzer](#)
- Also contributed to  [rust-lang/rust](#),  [rust-lang/rust-clippy](#),  [rust-lang/rustup](#),  [rust-lang/rust-mode](#)
-  [llvm/llvm-project](#),  [clangd/vscode-clangd](#),  [google/autocxx](#),  [yuin/goldmark](#), [more projects on GitHub](#).

Skills

- *Programming Languages*: Not limited to specific language. Especially proficient in C, C++, Java, Rust, Python, JavaScript/TypeScript, Verilog/SystemVerilog. Comfortable with Ruby, Swift, OCaml, Coq, Haskell, Agda, etc.
- *PL Theory*: Familiar with **type systems** (e.g. Hindley-Milner), formal verification and theory of computation.
- *Compilers / Virtual Machines*: **4 YoE**. Proficient in full compiler pipeline development:
 - Familiar with various **IRs** (e.g., SSA, MLIR, DBI, ANF, etc.), **optimizations** (e.g., Mem2Reg, GVN/GCM, register allocation, instruction scheduling, auto-vectorization, etc.), and **GC algorithms** (Mark-Sweep, Copying, Mark-Copy, etc.);
 - Knowledgeable about LLVM and LLVM-IR; read through parts of LLVM optimizations, familiar with LLVM's codegen module and related optimizations;
 - Understanding of NVIDIA GPU compiler architecture and optimizations, familiar with NVVM IR and PTX instructions.
- *IDE Development*: **2 YoE**. Familiar with IDE architecture based on **incremental computation**, especially the architecture and implementation of rust-analyzer and clangd; knowledgeable about plugin development for mainstream editors like VS Code and Emacs, and proficient with the **Language Server Protocol** specification and implementation.
- *Program Analysis*: **2 YoE**. Familiar with common static analysis algorithms (e.g., dataflow analysis, interval analysis, IFDS, pointer analysis with different sensitivities, etc.).
- *System Programming*: Familiar with computer architecture and operating systems, capable of assembly-level development and debugging, knowledgeable about Docker, GDB, CMake, and other tools.
- *Development Environment*: Proficient in Emacs; comfortable working in macOS and Linux; skilled in leveraging AI tools.

Misc

- **Languages**: Chinese (native), English (fluent)
- **Teaching Assistant**: *Programming in Practice* (Fall 2020), *Object-oriented Design and Construction* (S.T.A.R. responsible for TA work and system maintenance / Fall 2021, Spring 2022), *Principles and Techniques of Compilers* (Spring 2024).