roife

[redacted] | roifewu@gmail.com | roife | roife.github.io

教育背景

南京大学 2023.09 - 2026.06（预计）

硕士，计算机科学与技术 | PASCAL Lab，导师：李[redacted] | 研究方向为**程序语言**与**程序分析**。
助教工作：编译原理（2024 春）

北京航空航天大学 2019.09 - 2023.06

本科，计算机科学与技术 | GPA: 3.84/4.00；获**国家奖学金** / **优秀毕业生** / 2021 年计算机系统能力大赛编译赛一等奖等奖项。
助教工作：程序设计基础（2020 秋），面向对象设计与构建（2021 秋，2022 春）

工作经历

NVIDIA 2025.02 - 至今

OCG(Optimizing Code Generator) team GPU 编译器 LLVM 后端实习生

- 主导了 NVIDIA GPU 图形编译器与 NVVM 的访存指令 **vectorizer** 的合并，旨在对齐 LLVM 上游并降低维护开销：
  - 为了支持图形访存指令，独立设计了**多地址图形访存指令**的编码方案，以复用 LLVM 访存向量化器的核心流程；
  - 为图形指令添加 alias analysis 的支持，深入分析 BasicAA 和 SCEVAA 以支持新 intrinsics；
  - 改进了 LLVM 的 alignment inference 算法，通过对 SCEV 求解最大公约数，以支持对循环迭代变量的对齐推断；
  - 实现了**十余个**图形访存指令向量化相关的**优化**, 包括**非规则访存序列填充**、支持**整数地址向量化**等, 将向量化成功率提升了 **40%**；
- 参与图形编译器维护，修复了数个 Vectorization, SCEV 等优化的 bug；

Rust Foundation Fellowship Program 2024.09 - 2025.09

Rust 基金会开源资助（全球约 20 人） Project Fellow

- 作为 **Rust-lang Organization 成员**（rust-analyzer-contributors-team） 和 **rust-analyzer 维护者**（Rust 语言官方 IDE）之一，社区中贡献排名在**前 1%**，参与 issues 处理、PR 审核等维护工作；同时参与维护 rust 语言社区其他项目，如 rust-clippy 等；
- 实现了控制流高亮、快照测试更新等多项功能，并参与了大量 bug 修复，增强了 IDE 在代码理解、自动生成等多方面的能力；
- 为项目的 unicode 断字断行模块编写了 NEON 下的 **SIMD** 实现，使得该模块在 ARM 平台上提速 **6.5 倍**；
- v0.3.1992 **事故救火**：在发布新版本 4 小时后，社区发现该版本存在导致资源耗尽且无法结束进程的恶性 BUG。本人在 **3 小时内**定位到错误的依赖图搜索算法，并**设计新算法**解决了问题。该紧急修复控制了事故影响范围，避免了影响全球 Rust 开发者的工作。

个人项目

Vizsla roife/**vizsla** (WIP)

面向芯片前端设计的现代化 IDE · 硕士毕设项目 Rust / SystemVerilog

- 实现了一套面向可综合 SystemVerilog 的**语义分析框架**以及 IDE 基础设施，旨在为芯片设计配备现代 IDE 功能；
- 基于**增量计算**架构，设计并实现了一套增量分析 IR 和增量分析 pass，使得代码分析器无需全量更新即可得到准确的分析结果；
- 项目在功能、性能等指标上均达到**业界先进水平**：已完成面向 SystemVerilog 的代码导航、语义重构、代码补全、诊断等**数十项**现代 IDE 特性，并能够利用增量语义分析在各项功能上做到**毫秒级**延迟；基于语言服务器协议，适配 VS Code、Emacs 等主流编辑器；

Ailurus roife/**ailurus**

编程语言及工具链设计探索 · 个人兴趣项目 Rust

- 基于 **Martin-Löf 类型论**；支持 **dependent type**、dependent pattern matching、inductive datatype 等特性。实现了 propositional equality，使用 Normalization by Evaluation 进行等价检查，可实现简单的定理证明；
- 采用基于 **typeclass** 的 ad-hoc polymorphism，并基于此实现了**运算符重载**；实现了 **module system**，支持代码的命名空间和封装；
- 旨在作为实验平台，探索现代编程语言工具链（如编译器、IDE 等）的协同设计架构，提高编程语言开发的效率和可维护性。

Ayame No-SF-Work/**ayame**

SysY（C 子集）到 ARMv7 的编译器 · 全国大学生系统能力竞赛（编译器设计）比赛项目 Java / LLVM-IR / ARM

- 合作项目，个人主要负责编写面向 Machine IR 和体系结构的后端优化和代码生成，完成了基于图着色的**迭代寄存器合并**算法、**指令调度**、**死代码删除**、窥孔优化等，同时参与了部分块的编写；
- 同时负责项目的测试和 DevOps，利用 docker 和 GitLab CI 搭建了测试流程，并编写了 Python 脚本自动分析测试结果；

- 项目从零开始，完成了从语法解析到代码生成的完整编译器 pipeline，编写了大量 SSA IR 与 Machine IR 上的优化，最终在比赛中获一等奖。本项目在比赛中总排名第二，在**近一半样例上排名第一**，并在 1/3 的样例上优化效果超越 `gcc -O3` 与 `clang -O3`。

开源社区贡献

- 负责维护官方 IDE（语言服务器）rust-lang/**rust-analyzer**；在 rust 社区也贡献过 rust-lang/**rust**, rust-lang/**rust-clippy**, rust-lang/**rustup**，rust-lang/**rust-mode** 等项目；
- llvm/**llvm-project**，clangd/**vscode-clangd**，MikePopoloski/**slang**, google/**autocxx**，yuin/**goldmark**, moon-bitlang/**tree-sitter-moonbit**，更多项目见 GitHub。

专业技能

编程语言 能力不局限于特定编程语言。熟悉 C, C++, Rust, Java, Python, JavsScript/TypeScript, Verilog/SystemVerilog, EmacsLisp；学习并使用过 Ruby, Swift, OCaml, Haskell, Coq, Agda 等；

程序语言理论
- 形式语义、类型论、计算模型、自动机等基础理论；学习过 Coq、Agda 等定理证明器的使用；
- （**类型系统**）Hindley-Milner, Subtyping, System F, Dependent Type 等类型系统的理论和实现；
- （**静态分析**）数据流分析、控制流分析、IFDS、采用不同敏感度的**指针分析**等常用分析算法

编译器设计 **3 年经验**。精通编译器从语法解析到代码生成的**全 pipeline 开发**，熟悉多种 **IR**（ANF, SSA, CPS 等）：
- （**语言实现**）面向对象、函数式等多种范式语言的编译过程，以及双向类型检查等语言特性的实现；
- （**IDE 开发**） **2 年经验**，基于**增量计算**和 **LSP** 的 IDE 架构，编辑器插件开发；熟悉 rust-analyzer 和 clangd；
- （**编译优化**）编译器中端、后端的分析和优化，包括 Mem2Reg, GVN, RegAlloc, List Scheduling 等；熟悉 **LLVM** 上的分析优化实现和代码库，熟悉 LLVM-IR 与 MLIR；

高性能计算
- X86 和 ARM 指令集架构；了解超标量处理器的架构和现代存储架构；了解 NVIDIA GPU 的架构和渲染管线；
- （**性能分析**）NVIDIA Nsight, Intel VTune Profiler 等性能分析工具的使用；
- （**优化加速**）了解 **OpenMP**, **CUDA** 等并行计算模型；SSE, AVX, NEON 等常见的 SIMD 架构及应用

应用开发 Ruby on Rails, Django, SwiftUI 等开发框架；PostgreSQL、Redis 等数据库；Docker 和 CI/CD 配置等 DevOps 工作；

开发环境 熟悉 Emacs / VS Code，习惯在 macOS / Linux 下工作；熟练使用生成式 AI 工具（如 GitHub Copilot）提高效率。

其他

- **Talks**:
  - *Exploring rust-analyzer: from Incremental Computation to Modern IDE* (RustChinaConf 2025)