

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█ 研究方向为程序语言、程序分析与 HDL	
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	

工作经历

Rust Foundation Fellowship Program	2024.09 - 2025.06
• 维护 rust-analyzer (Rust 官方 IDE) <ul style="list-style-type: none">社区贡献排名 23/972, 解决超过 50 个 issues; 参与维护语义分析、类型检查等多个模块, 提高语义分析的正确性和项目的稳健性; 并为项目添加了多项新功能, 如控制流导航、泛型约束展示等;为 rust-analyzer 的断行算法编写了 NEON 下的 SIMD 实现, 使得该模块在 ARM 平台上提速 6.5 倍;解决 v0.3.1992 版本 P0 事故: 新版本发布 4 小时后, 社区发现该版本存在恶性 BUG, 会导致设备计算资源耗尽且无法结束进程。本人根据反馈在 3 小时内定位到错误算法逻辑, 设计了新算法修复了问题并重新发版。本次紧急修复有效控制了事故影响范围, 避免影响全球范围内 Rust 使用者的开发工作, 提升了项目稳健性。	
• 开源社区日常维护: 参与社区日常维护工作, 包括会议讨论, BUG 修复, PR 审核等。	

奖项荣誉

- 2022 年国家奖学金 (学年专业排名 1/195), 北京航空航天大学优秀毕业生
- 2021 年全国大学生计算机系统能力大赛·编译系统设计赛 (华为毕昇杯) 一等奖, 排名第二
- 蓝桥杯 C++ 程序设计竞赛北京赛区一等奖, 国赛三等奖
- 另获其他各类省奖与校级学业、竞赛奖学金十余次

项目开发

Vizsla · 面向芯片开发的现代化 Verilog/SV IDE (Rust / SystemVerilog)	(开发中)
• (主要负责人) 负责设计并编写了 IDE 的核心架构和增量计算流程, 并完成了大部分的功能开发;	
• 项目旨在为芯片设计配备现代 IDE 功能, 完成了代码导航、补全等现代 IDE 特性, 以提升编码效率和代码质量;	
• 基于 LSP 协议和增量计算架构, 实现了增量语义分析框架。	
LLVM-Lite · 面向深度学习神经网络算子的轻量级端侧编译器 (C++ / LLVM)	🌐 roife/llvm-lite
• (独立设计实现) 华为研究课题, 作为本科毕业设计获得优秀评价;	
• 本课题旨在利用端侧推理设备上神经网络的形状信息, 对深度学习算子进行二次优化, 以减少算子运行时的时空开销;	
• 独立完成了一个轻量级的端侧编译器用于端侧算子的二次优化, 并完成了 LLVM 代码生成模块的裁剪工作;	
• 成功将测试样例中的深度学习算子的运行时间降低了 6%, 并将目标文件大小降低了 38%。	
Ayame · 毕昇杯比赛项目, 从 C 子集到 LLVM-IR/ARMv7 编译器 (Java / LLVM / ARMv7)	🌐 No-SF-Work/ayame
• (合作开发) 负责完成了图着色寄存器分配算法以及面向体系结构的后端优化, 并负责了本地 CI 和评测系统的搭建;	
• 该项目在比赛中近一半样例中排名第一, 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

- Rust-lang Member, rust-analyzer contributors team: 工作集中在 [rust-lang/rust-analyzer](#), 同时也贡献过 [rust-lang/rust-clippy](#), [rust-lang/rustup](#), [rust-lang/rust-mode](#) ;
- [llvm/llvm-project](#), [clangd/vscode-clangd](#), [google/autocxx](#), [yuin/goldmark](#), 更多项目见 [GitHub](#)

专业技能

编程语言	不限于特定语言。熟悉 C, C++, Rust, Java, Python, Verilog; 较熟悉 JS, Ruby, Swift, OCaml, Coq, Haskell 等。
程序语言理论	了解形式语义、形式语言、形式化验证和计算理论的相关知识, 熟悉类型系统的理论和实现。
编译器 / IDE	4 年开发经验。熟悉编译优化和多种中间代码表示 (如 SSA, CPS 等); 对 LLVM 有一定了解; 熟悉基于 LSP 协议和基于增量计算的 IDE 架构, 尤其熟悉 rust-analyzer。
程序分析	2 年开发经验。熟悉常见静态分析算法 (如数据流分析、指针分析、IFDS 等)。
系统编程	了解体系结构和操作系统底层知识, 能进行底层的开发调试工作, 熟悉 Docker、CMake 等常用工具。
开发环境	熟悉 Emacs; 习惯在 macOS 和 Linux 环境下工作; 能熟练使用 AI 提高工作效率。

其他

- 外语: 英语 (CET-6);
- 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), 编译原理 (2024 春)。