

Yuyang (Peter) RONG

+1 530 601 3646 ◇ PeterRong96@gmail.com ◇ [Website](#) ◇ [GitHub](#)

EDUCATION

UC Davis

Sep 2019 - Jun 2024

Ph.D. candidate in Computer Science

Davis, CA

- Research interests: Software security, compiler validation, LLM in software engineering
- Skills: **C/C++/Rust (9/10), Python (8/10), LLVM (IR, CodeGen)**, GitHub, Jira, Docker, German
- Community impacts: **LLVM contributor**. Bug fixes in backends, e.g. X86, AArch64, etc.
- Leadership & project management: Managing six students and three projects in parallel

ShanghaiTech University

Sep 2015 - Jun 2019

B.E. Computer Science and Technology

Shanghai, China

- GPA 3.79/4 (Rank: 5/124) Excellent Undergraduate of Shanghai (2019) Scholarship of Shanghai (2016)

EXPERIENCE

Meta Platforms Inc.

Jul 2024 - Current

Research Scientist

Menlo Park, CA

- Focused on mobile compiler optimization research and implementation
- Build optimization to reduce size of all Meta's apps (Facebook, Instagram, etc.) running on iOS platform.
- Directly contributing to open-source projects including LLVM.

Advanced Micro Devices, Inc. (AMD)

Jul 2023 - Dec 2023

Research Intern

San Jose, CA

- Focused on developing a scheduling algorithm based on reinforcement learning for AI Engine (AIE).
- Preliminary results show that the new algorithm can reduce 5% cycles in the generated assembly code.
- Implemented GitHub Action script for weekly fuzzing of AIE, cooperated with DevOps to deploy the fuzzer.

Advanced Micro Devices, Inc. (AMD)

Jun 2022 - Dec 2022

Research Intern

San Jose, CA

- Focused on testing compiler backend (CodeGen) of AI Engine (AIE).
- Implemented and open-sourced [IRFuzzer](#) in 2000 LoC C++ to accommodate for the compiler infrastructure.
- Found over 40 missing features in AIE. Found **74 confirmed bugs in LLVM**, **44 fixed**, [bug tracker](#).
- Lightning talk accepted to 2022 LLVM Developer's Meeting, [recording](#).

Bytedance Ltd.

Jun 2020 - Sep 2020

Research Intern

Mountain View, CA

- Focused on optimizing fuzzer Angora's gradient solver and alleviate its branch collision problem.
- Implemented a fuzzer [Valkyrie](#) with a runtime in ~2000 LoC in C++ and a gradient solver in ~3000 LoC in Rust.
- Found six bugs in open-source libraries, improved branch coverage by 41% compared to Angora.

SELECTED PUBLICATIONS

IrFuzzer: Specialized Fuzzing for LLVM Backend Code Generation

2025

International Conference on Software Engineering (ICSE)

Under peer review

UniTSyn: A Large-Scale Dataset Capable of Enhancing the Prowess of Large Language Models for Program Testing

2024

International Symposium on Software Testing and Analysis (ISSTA)

Understanding Programs by Exploiting Fuzzing Test Cases

2023

Association for Computational Linguistics (ACL)

Valkyrie: Improving Fuzzing Performance Through Principled Techniques

2022

Software Quality, Reliability, and Security (QRS)

(Best paper award)

IntEgrity: Finding Integer Errors by Targeted Fuzzing

2020

Security and Privacy in Communication Networks (SecureComm)

TEACHING

ECS153: Computer Security

(Best TA award) Spring 2023

ECS032A: Introduction to Programming

Fall 2020