Benjamin Steenhoek, PhD

benjijang.com · benjaminjsteenhoek@gmail.com · Scholar · github.com/bstee615 · linkedin.com/in/ben-steenhoek

AI/ML researcher building code-editing and agentic systems for software engineering. Current focus areas are LLM post-training and agent harness engineering. Research interests include agent evaluation, LLM fine-tuning, software security, and program analysis.
Based in Des Moines, Iowa.

EXPERIENCE

Senior Researcher — **Fall 2024–Present**
Microsoft CodeAI · Des Moines, IA

- Drove research engineering for **GitHub Copilot's primary agentic evaluation platform**, building its scalable harness environment & tooling and onboarding key benchmarks. During my tenure, we grew it from a **greenfield prototype to 1M+ monthly executions**, and it now powers evaluations for flagship coding agents, including VS Code Agent, Copilot Cloud Agent, and Copilot CLI.
- Led post-training, evaluation, and rollout of a production model for GitHub Copilot Inline Suggestions that **increased suggestion acceptance by 10%** and enabled a configurable eagerness feature that **addressed 10+ long-standing community issues**.

Research Intern — **May 2023–Aug 2024**
Microsoft CodeAI · Des Moines, IA

- Conducted a user study on in-IDE AI-powered security vulnerability detection & repair with **17 professional developers across 24 projects, 6.9K files, and 1.7M+ lines of code**; *Closing the Gap*, ICSE 2025 (21% acceptance rate).
- **Improved unit-test readability and maintainability by up to 21%** through reinforcement learning; the fine-tuned Codex model was **less than half the cost of GPT-4 and outperformed it on 4 of 7 quality metrics**; *Reinforcement Learning from Automatic Feedback*, DeepTest @ ICSE 2025.

Research Assistant — **2020–2024**
Program Analysis & AI Lab · Iowa State University · Ames, IA

- Achieved **state-of-the-art Big-Vul vulnerability detection performance (96.46 F1)** by combining a dataflow-inspired graph model with an LLM. Our model trained in 9 minutes—**75× faster than the strongest baseline**—and detected 8.7 of 17 real-world vulnerabilities where baselines found none; *DeepDFA*, ICSE 2024 (code; 22% acceptance rate).
- Built a C/C++ execution tracing engine from scratch for *TRACED*, ICSE 2024 (code; 22% acceptance rate), enabling execution-aware pre-training that **improved complete path prediction by 12.4% and runtime value prediction by 25.2%**. Built a Java counterpart later used for *CodeSense*, ICLR 2026.
- Reproduced and evaluated **9 state-of-the-art vulnerability detection models across 2 datasets**, uncovering substantial run-to-run variance and low agreement among model outputs; *An Empirical Study of Deep Learning Models for Vulnerability Detection*, ICSE 2023 (code; 26% acceptance rate).
- Collaborated with Columbia University's ARiSE Lab and Carnegie Mellon University's CERT division; built open-source static and dynamic analysis tools including tree-climber and pal-tools.

SELECTED PUBLICATIONS

See all: scholar.google.com/citations?user=uJk56S4AAAAJ

1. Monoshi Kumar Roy, Simin Chen, **Benjamin Steenhoek**, Jinjun Peng, Gail Kaiser, Baishakhi Ray, and Wei Le. (2026). *CodeSense: a Real-World Benchmark and Dataset for Code Semantic Reasoning. ICLR'26*. Paper · Code
2. Spandan Garg, **Benjamin Steenhoek**, and Yufan Huang. (2026). *Saving SWE-Bench: A Benchmark Mutation Approach for Realistic Agent Evaluation. CAIN'26*. Paper · Code
3. **Benjamin Steenhoek**, Kalpathy Sivaraman, Renata Saldivar Gonzalez, Yevhen Mohylevskyy, Roshanak Zilouchian Moghaddam, and Wei Le. (2025). *Closing the Gap: A User Study on the Real-world Usefulness of AI-powered Vulnerability Detection & Repair in the IDE. ICSE'25*. Paper
4. **Benjamin Steenhoek**, Michele Tufano, Neel Sundaresan, and Alexey Svyatkovskiy. (2025). *Reinforcement Learning from Automatic Feedback for High-Quality Unit Test Generation. DeepTest @ ICSE'25*. Paper
5. **Benjamin Steenhoek**, Md Mahbubur Rahman, Monoshi Kumar Roy, Mirza Sanjida Alam, Hengbo Tong, Swarna Das, Earl T. Barr, and Wei Le. (2024). *To Err is Machine: Vulnerability Detection Challenges LLM Reasoning. arXiv*. Paper
6. Yangruibo Ding, **Benjamin Steenhoek**, Kexin Pei, Gail Kaiser, Wei Le, and Baishakhi Ray. (2024). *TRACED: Execution-aware Pre-training for Source Code. ICSE'24*. Paper · Code
7. **Benjamin Steenhoek**, Hongyang Gao, and Wei Le. (2024). *Dataflow Analysis-Inspired Deep Learning for Efficient Vulnerability Detection. ICSE'24*. Paper · Code
8. **Benjamin Steenhoek**, Md Mahbubur Rahman, Shaila Sharmin, and Wei Le. (2023). *Do Language Models Learn Semantics of Code? A Case Study in Vulnerability Detection. arXiv*. Paper
9. **Benjamin Steenhoek**, Md Mahbubur Rahman, Richard Jiles, and Wei Le. (2023). *An Empirical Study of Deep Learning Models for Vulnerability Detection. ICSE'23*. Paper · Code

10. Ashwin Kallingal Joshy, Xueyuan Chen, **Benjamin Steenhoek**, and Wei Le. (2021). *Validating Static Warnings via Testing Code Fragments. ISSTA'21*. Paper

EDUCATION

PhD, Computer Science — Iowa State University — 2019–2024
Dissertation: *Understanding and improving deep learning models for vulnerability detection*

MS, Computer Science — Iowa State University · GPA 3.91/4.00 — 2019–2021
Thesis: *Refactoring programs to improve the performance of deep learning for vulnerability detection*

BS, Computer Science — Bob Jones University · *magna cum laude* · GPA 3.84/4.00 — 2016–2019

INVITED TALKS & SERVICE

- **Reviewer** · ACM TOSEM — **2025, 2026**
- **Reviewer** · Empirical Software Engineering — **2024, 2026**
- **Program Committee** · SVM — **2026**
- **Program Committee** · FSE'25 IVR — **2025**
- **Program Committee** · FORGE — **2025**
- **Invited Talk** · IEEE DISTILL — **2025**
- **Reviewer** · IEEE TIFS — **2023**

SELECTED PROJECTS

See all: github.com/bstee615

- **DeepDFA** — efficient, dataflow-inspired vulnerability detection
- **TRACED** — C/C++ execution tracing for model pre-training
- **cfactor** — policy-driven refactoring for C programs
- **tree-climber** — program analysis tools for C built on tree-sitter
- **pal-tools** — dynamic analysis and code-generation utilities
- **rrun** — Git-aware remote command runner over SSH
- **wslwatch** — watchdog that keeps WSL2 distributions running