Ziyang Li

https://liby99.github.io/

Email : liby99@seas.upenn.edu
Mobile : +1-858-699-3237

Education

- **University of Pennsylvania** — Philadelphia, PA
  Ph.D. Computer and Information Science, Advisor: Mayur Naik; GPA: 4.0 — *Jul 2019 – Present*
- **University of California – San Diego** — La Jolla, CA
  B.S. Computer Science (3.9/4.0); B.S. Mathematics (3.7/4.0); GPA: 3.6 — *Sep 2015 – Jun 2019*

Research Interest

My research focuses on combining program analysis, machine learning, and symbolic reasoning. I design methods and tools with a goal of finding security vulnerabilities, fixing bugs, improving code quality, and helping developers write elegant, safe and performant code.

Publications

- **ARBITRAR: User-Guided API Misuse Detection**
  Ziyang Li, Aravind Machiry, Binghong Chen, Mayur Naik, Ke Wang, Le Song
  IEEE Security and Privacy (S&P) 2021. [Paper] [Code] [Talk]
- **HOPPITY: Learning Graph Transformations to Detect and Fix Bugs in Programs**
  Elizabeth Dinella, Hanjun Dai, Ziyang Li, Mayur Naik, Le Song, Ke Wang
  International Conference on Learning Representations (ICLR) 2020, **Spotlight**. [Paper]

Preprints:

- **Scallop: From Probabilistic Deductive Databases to Scalable Differentiable Reasoning**
  Jiani Huang, Ziyang Li, Binghong Chen, Karan Samel, Mayur Naik, Le Song, Xujie Si
  Under review at NeurIPS 2021

Research Experiences

- **Research Assistant, UPenn PEARL** — University of Pennsylvania
  Advisor: Mayur Naik — *July 2019 – Present*
  - **Probabilistic DataLog Solver**: We build a scalable and flexible probabilistic DataLog engine, oriented towards machine learning applications. Comparing to existing tools like ProbLog, we support top-k provenance tracking, type inference, and aggressive compiler optimization.
  - **Find API Misuses with User Interaction**: We build Arbitrar, an Active Learning based method to involve human-in-the-loop to find API Misuses bugs in large code corpus. We demonstrated that we can find the bug given a target API with only a few rounds of user interactions, where we learn the correct usage from scratch. We showed that Arbitrar is much more effective than existing tools like APISan which has a huge amount of false positives.
  - **Learn to Find and Fix JavaScript Bugs**: We present Hoppity, a GNN based learning method to find and fix JavaScript bugs, trained on a huge dataset consists of Github commits. Hoppity correctly detects and fixes bugs in 9,490 out of 36,361 programs in an end-to-end fashion. Given the bug location and type of the fix, Hoppity also outperforms the baseline approach by a wide margin.
- **Undergraduate Research Assistant, UCSD PL** — Univerysity of California – San Diego
  Advisor: Sorin Lerner — *Sep 2018 – Jun 2019*
  - **Proverbot9001**: Learn to prove theorems in Coq automatically using machine learning and neural networks.
- **Undergraduate Research Assistant, UCSD VISCOMP** — Univerysity of California – San Diego
  Advisor: Ravi Ramamoorthi — *Mar 2018 – Jun 2019*

- **Spherical Harmonics**: I developed a Spherical Harmonics based real time realistic lighting for AR applications.

- **Undergraduate Research Intern, UCSD Design Lab** Univerysity of California – San Diego
  Advisor: Scott Klemmer *Jun 2017 – Jun 2018*
  - **Galileo**: We built a platform for people to design and run life-style related experiments with community. We teach users to design and conduct double blind experiments, recruit experiment candidates, and generate experiment reports.

Working Experiences

- **Visa, Inc.** *Research Intern, Mentor: Ke Wang* Virtual, *May 2020 – July 2020*
- **Coursera, Inc.** *Front-end Engineer Intern* Mountain View, CA, *Jun 2018 - Sep 2018*
- **Deep Media, Ltd.** *Full-stack Engineer Intern* Shenzhen, China, *Sep 2016 – Jan 2017*
- **Yobs Technology** *Full-stack Engineer Intern* Los Angeles, CA, *Jan 2016 – Sep 2016*
- **Easyhin** *Front-end Engineer Intern* Shenzhen, China, *Aug 2015 – Sep 2015*

Fellowships

- **KPCB Fellows 2018** *Engineering Fellows* San Francisco, *June 2018*

Teaching Experiences

- **Teaching Assistant** *CIS 547, Software Analysis* University of Pennsylvania, *Aug 2020 – Dec 2020*
- **Tutor** *CSE 190, Virtual Reality Technology* University of California – San Diego, *Mar 2019 – Jun 2019*
- **Tutor** *CSE 165, 3D User Interaction* University of California – San Diego, *Jan 2019 – Mar 2019*
- **Tutor** *CSE 130, Programming Language* University of California – San Diego, *Sep 2018 – Dec 2018*
- **Tutor** *CSE 163, Advanced Computer Graphics* University of California – San Diego, *Mar 2018 – Jun 2018*
- **Tutor** *CSE 167, Intro to Computer Graphics* University of California – San Diego, *Jan 2018 – Mar 2018*
- **Tutor** *CSE 12, Data Structure* University of California – San Diego, *Jan 2017 – Mar 2017*

Selected Projects

- **Probabilistic DataLog Engine**: A probabilistic datalog engine with high performance optimizations oriented towards machine learning applications. Written in Rust.
- **Under-constrained Symbolic Execution Engine**: High performance under-constrained symbolic execution engine for LLVM IR written in Rust. Used in Arbitrar.
- **LLVM IR Binding for Rust**: Safe LLVM Binding for Rust. Used in Arbitrar. [Github]
- **AoSoA Storage**: Array-of-struct-of-array storage system for high performance parallel computing with Kokkos and Cabana. Oriented for physics simulation applications. Used by UPenn CG Group. [Github]
- **MPM-RS**: Material point method written in Rust. [Github]
- **Geometry Sketchpad**: Geometry sketching GUI application written in Rust. [Github]
- **Menhera**: A TypeScript-like functional programming language compiler written in OCaml. [Github]
- **Fourier Depth of Field**: Fourier transform based depth of field analysis for RayTracer. [Github]
- **Neon Ping Pong**: VR Ping Pong Game written in C++. [Website] [Video]
- **Space Escape**: VR Room Escape Puzzle Game settled in Space Station. Developed in Unity. [Website] [Video]
- **Rotamina**: Character animator and simulator with GUI. Written in C++. [Github]

Skills

- **Languages**: Rust, C++/C, TypeScript/JavaScript, C#, Python, OCaml, Java, Haskell, Coq, SQL
- **Libraries/Tools**: PyTorch, Unity, Rocket, React, ExpressJs, Asp.net
- **Design**: Adobe Photoshop, Premiere, Illustrator, Blender, Cinema 4D

Last update: Jul 31, 2021