Shuyang Liu

shuyang-liu.github.io liushuyang860@gmail.com

EDUCATION

Huazhong University of Science and Technology **GPA: 3.92/4.00**
B.E. in Computer Science *Sept. 2020 - Jun. 2024*

PUBLICATION

Symmetry-Preserving Program Representations for Learning Code Semantics
Kexin Pei, Weichen Li*, Qirui Jin*, **Shuyang Liu**, Scott Geng, Lorenzo Cavallaro, Junfeng Yang, Suman Jana
Accepted by Symposium on Machine Programming (MAPS), 2023. (workshop version)
Submitted to International Conference on Learning Representations (ICLR), 2024. Under Review.

RESEARCH EXPERIENCE

Exploiting Code Symmetries for Learning Program Semantics **Advisor: Suman Jana**
Columbia University *Mar. 2023 - Aug. 2023*

- Employed Tree-sitter to construct Program Dependence Graphs (PDGs) based on data and control dependencies between statements.
- Implemented nine types of semantic-preserving source transformations, including Statement Permutation, in which I extended existing work beyond only two-statement permutation to all possible PDG automorphisms.
- Evaluated a range of traditional and LLM baseline models for method name prediction.

Enhancing Code Semantics Learning with Fine-Grained PDGs **Advisor: Kexin Pei**
University of Chicago *Sept. 2023 - Present*

Extended our existing work from inter-statement analysis to intra-statement analysis, constructing fine-grained Program Interpretation Graphs for the model to learn code semantics.

- Static typed language: Employed JavaParser to extract dependencies between tokens for Java.
- Dynamic typed language: Applied PyType to perform static type inference for Python.

Automatic identification of Bug Inducing Commits **Advisor: Ming Wen**
CASTLE Lab, Hong Kong University of Science and Technology *Ongoing*

- Systematically validated bug-fixing and associated bug-inducing commits for 237 bugs across five large open-source Java projects. Employed bisection method for precise identification.
- Reproduced SZZ Unleashed algorithm using Defects4J database, establishing a foundational baseline for performance benchmark.
- Reproduced "Reducing the Search Space of BICs using Failure Coverage" approach. Expanded the research by intersecting its results with those of the SZZ Unleashed.

TECHNICAL SKILLS

- **Languages**: C/C++, Java, Python (Pytorch, Tensorflow)
- **Tools**: Git, Linux, LaTeX, CodeQL, JavaParser, and Tree-sitter
- **Areas of Interest**: Software Engineering, Security, Compilers, and Machine Learning