

# Zhaoyang Zhang

Hong Kong | zhaoyang.zhang@connect.ust.hk | +86 17320639795 | hanquanjushi.github.io  
linkedin.com/in/hanquanjushi | github.com/hanquanjushi

## Education

- |  |                     |
|--|---------------------|
| <b>The Hong Kong University of Science and Technology, Hong Kong</b>     | Aug 2025 – Present  |
| • Ph.D in Computer Science and Engineering (Advisor Jiasi Shen)          |                     |
| <b>University of Science and Technology of China, Anhui, China</b>       | Sep 2021 – Jul 2025 |
| • Bachelor of Engineering in Computer Science and Technology             |                     |
| • GPA (overall): 3.89/4.3 (90.36/100), ranking: 14/258 ( <b>top 5%</b> ) |                     |

## Internship

- |  |                     |
|--|---------------------|
| • HKUST Summer Undergraduate Visiting Internship Program | Jul 2024 – Aug 2024 |
|--|---------------------|

## Peer Reviewed Publications

Shangyu Li, **Zhaoyang Zhang**, Sizhe Zhong, Diyu Zhou, Jiasi Shen. “**A Sound Static Analysis Approach to I/O API Migration.**” The ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA’25)

## Projects

### Autonomous Driving Rare Scene Construction System

- Developed SceneGen, an innovative framework to enhance safety verification of Automatic Emergency Braking (AEB) systems in autonomous driving. Leveraged large language models and prompt engineering to convert public accident reports descriptions into Scenic scripts within the NL2Scenic framework, surpassing the ScenicNL baseline by 10%. Generated the AEB-Scenario dataset, enabling simulation in the Carla platform and producing multi-modal data in nuScenes format, including RGB images, radar point clouds, GPS/IMU data, and HD maps, supporting perception model training, strategy verification, and safety evaluation in complex traffic environments.

### Compiler

github.com/SysY-Compiler

- This project built a complete SysY (a subset of the C language) compiler and implemented intermediate code optimization. Initially, a lexical and syntax analyzer based on Flex and Bison was implemented, followed by the conversion from syntax tree to LLVM intermediate code. The intermediate code was then translated into Loongson assembly instructions. Finally, optimizations were applied to the intermediate code using optimizing passes. After constructing the dominator tree, it used Mem2Reg (inserting phi functions and accomplishing variable renaming) to eliminate memory operations and perform dead code elimination.

## Teaching

2026 Teaching Assistant, COMP 4321 Search Engines for Web and Enterprise Data (Spring), HKUST

## Skills

- Adept in C/C++, Python, Verilog
- Proficient in LLVM, RISC-V

## English

- **IELTS** (2024): 7.0
- **GRE General Test** (2022): Verbal 158 (78%), Quantitative 170 (96%)