Hyunjae Suh

Irvine, CA | hyunjas@uci.edu | Website | Linkedin | Google Scholar

Summary

Ph.D. student in Software Engineering specializing in **Generative AI and Machine Learning for software development**. Experienced in evaluating and improving LLM-generated code through advanced prompting strategies, empirical studies, and ML-based risk prediction for large-scale software systems.

Education

University of California, Irvine Sep. 2023 – Present
Ph.D. in Software Engineering *Irvine, CA*

Kookmin University Mar. 2017 – Aug. 2023
B.S. in Computer Science *Seoul, South Korea*

Industry Experience

Applied Scientist Intern Jun. 2025 – Sep. 2025
Amazon *Austin, TX*

- Developed ML-based change risk models on large-scale Prime Video codebases to improve deployment reliability by proactively flagging risky commits.
- Engineered diff-aware representations using LLMs and evaluated ensemble classifiers (Random Forest, XGBoost), reaching 84.60 F1-score and demonstrating the effectiveness of feature design in risk prediction tasks.
- Conducted large-scale experiments analyzing predictive features of risky changes, integrating risk scores into deployment pipelines to reduce failed releases.

Graduate Research Assistant Aug. 2023 – Dec. 2023
eBay *Remote*

- Fine-tuned open-source LLMs for commit message generation, reducing documentation effort and improving contextual relevance.
- Applied prompt engineering to standardize commit messages across repositories, improving clarity and consistency in documentation practices.
- Benchmarked multiple LLMs, analyzing trade-offs in accuracy, conciseness, and efficiency for engineering workflows.

Research Experience

Ph.D. Researcher – AI & Software Engineering Sep. 2023 – Present
University of California, Irvine *Irvine, CA*

- Conduct empirical studies on LLMs for software engineering tasks, with focus on accessibility, quality, and attribution.
- Built a detection pipeline combining fine-tuned LLMs, code embeddings, and feature-based classifiers, achieving state-of-the-art results (82.55 F1-score) on benchmark datasets — published at **ICSE 2025**.
- Designed and tested prompting strategies (Few-Shot, Self-Criticism, Multi-Agent Debate, ReAct) to reduce accessibility violations of AI-generated applications.
- Developed *FeedA11y*, a ReAct-based pipeline integrating accessibility evaluation feedback (IBM Equal Access, AChecker) into refinement loops for accessible code generation.
- Conducted large-scale experiments on fine-tuning order in LLMs for software engineering tasks, analyzing cross-task interference and transfer dynamics.

Publications

An Empirical Study on Automatically Detecting AI-Generated Source Code: How Far Are We?
***Hyunjae Suh**, Mahan Tafreshipour, Jiawei Li, Adithya Bhattiprolu, Iftekhar Ahmed.*
ICSE 2025 (47th IEEE/ACM International Conference on Software Engineering). [DOI]

Human or LLM? A Comparative Study on Accessible Code Generation Capability
***Hyunjae Suh**, Mahan Tafreshipour, Sam Malek, Iftekhar Ahmed.*
arXiv preprint, 2025.

Does the Order of Fine-tuning Matter and Why?
*Qihong Chen, Jiawei Li, **Hyunjae Suh**, Lianghao Jiang, Zheng Zhou, Jingze Chen, Jiri Gesi, Iftekhar Ahmed.*
arXiv preprint, 2024.

Projects

Detection of LLM-generated Source Code ICSE 2025

- Developed a **state-of-the-art detection pipeline** combining fine-tuned LLMs, code embeddings, and feature-based classifiers for automatic code attribution.
- Achieved **82.55 F1-score** on benchmark datasets, setting a new standard for detecting LLM-generated source code.

Accessible Code Generation via Prompting in LLMs

- Designed and implemented *FeedA11y*, a **ReAct-based pipeline** integrating formal accessibility evaluation tools (IBM Equal Access, AChecker) into LLM refinement loops.
- Evaluated advanced prompting strategies (Self-Criticism, Multi-Agent Debate) to improve WCAG 2.1 compliance in AI-generated UI code.

The Impact of Fine-tuning Order on LLMs for SE

- Conducted large-scale empirical experiments analyzing the effect of fine-tuning task order on cross-task interference and knowledge transfer across software engineering tasks (e.g., Code Translation, Code Summarization).

Service

Reviewer May 2025 – Present
ACM Transactions on Software Engineering and Methodology (TOSEM)

Reviewer Jun 2025 – Present
IEEE Transactions on Reliability (TR)

Teaching Experience

Teaching Assistant Sep. 2023 – Present
University of California, Irvine *Irvine, CA*

- IN4MATX 115 - Software Testing, Analysis, and Quality Assurance
- ICS 10 - How Computers Work
- ICS 32 - Programming with Software Libraries
- CS 121 - Information Retrieval

Technical Skills

Languages: Python, Java, R, JavaScript, C++, SQL
ML/Frameworks: PyTorch, Hugging Face, scikit-learn, Keras, Pandas, NumPy, Matplotlib, XGBoost, LangChain
Tools: AWS SageMaker, AWS Bedrock, AWS S3, Git, LaTeX
Code Analysis: Checkstyle, Detekt, ESLint, Soot, Understand
Research Methods: Empirical Studies, Quantitative Analysis, Data Visualization