STEVENS
INSTITUTE OF TECHNOLOGY
1870

Ph.D. DISSERTATION DEFENSE

Candidate:	Xing Qian
Degree:	Doctor of Philosophy
School/Department:	Schaefer School of Engineering and Science / Department of Systems Engineering
Date:	Monday, August 17, 2026
Time/Location:	9:00 a.m. ET / Zoom: https://stevens.zoom.us/j/93263851707
Title:	From Detection to Refactoring: Advancing Software Code Clone Management
Chairperson:	Dr. Eman Abdullah AlOmar, Department of Systems Engineering, Stevens Institute of Technology
Committee Members:	Dr. Lu Xiao, Associate Professor, Department of Systems Engineering, Stevens Institute of Technology Dr. Ying Wang, Associate Professor, Department of Systems Engineering, Stevens Institute of Technology Dr. Tegan Brennan, Assistant Professor, Department of Systems Engineering, Stevens Institute of Technology

ABSTRACT

Code clones are common in software systems because developers frequently reuse code through copy-and-paste. Although this practice can improve development efficiency, it also increases maintenance effort by requiring duplicated code to be updated consistently. Extract Method refactoring is a widely used technique for removing duplicated code while preserving program behavior. Recent advances in Large Language Models (LLMs) and AI coding agents have created new opportunities for automating software refactoring, but their effectiveness and practical applicability for code clone refactoring remain insufficiently understood.

This dissertation investigates the use of LLMs for code clone management through four complementary studies. The first study evaluates the ability of state-of-the-art LLMs to identify and refactor different types of code clones under controlled experimental settings. The second study analyzes clone-related pull requests in open-source software repositories to compare the behavior of AI coding agents with that of human developers in practical software development. Building on the findings of these studies, the third study proposes an LLM-driven workflow that integrates clone detection, Extract Method refactoring, structural validation, compilation checking, automated testing, and iterative feedback. The fourth study explores the use of LLMs for method name recommendation during Extract Method refactoring.

The experimental results indicate that modern LLMs perform well on straightforward clone refactoring tasks but encounter greater challenges when handling more complex scenarios involving cross-file dependencies, compilation constraints, and behavioral correctness. The pull request analysis shows that AI coding agents can effectively assist with well-scoped clone refactoring tasks while human developers remain important for broader architectural decisions. The proposed workflow improves the reliability of LLM-generated refactorings by incorporating multiple validation stages and iterative refinement. Together, these studies provide empirical evidence on the capabilities and limitations of LLMs for code clone management and present practical techniques for supporting automated clone refactoring in software development.