

# Towards Effective Validation and Integration of LLM-Generated Code

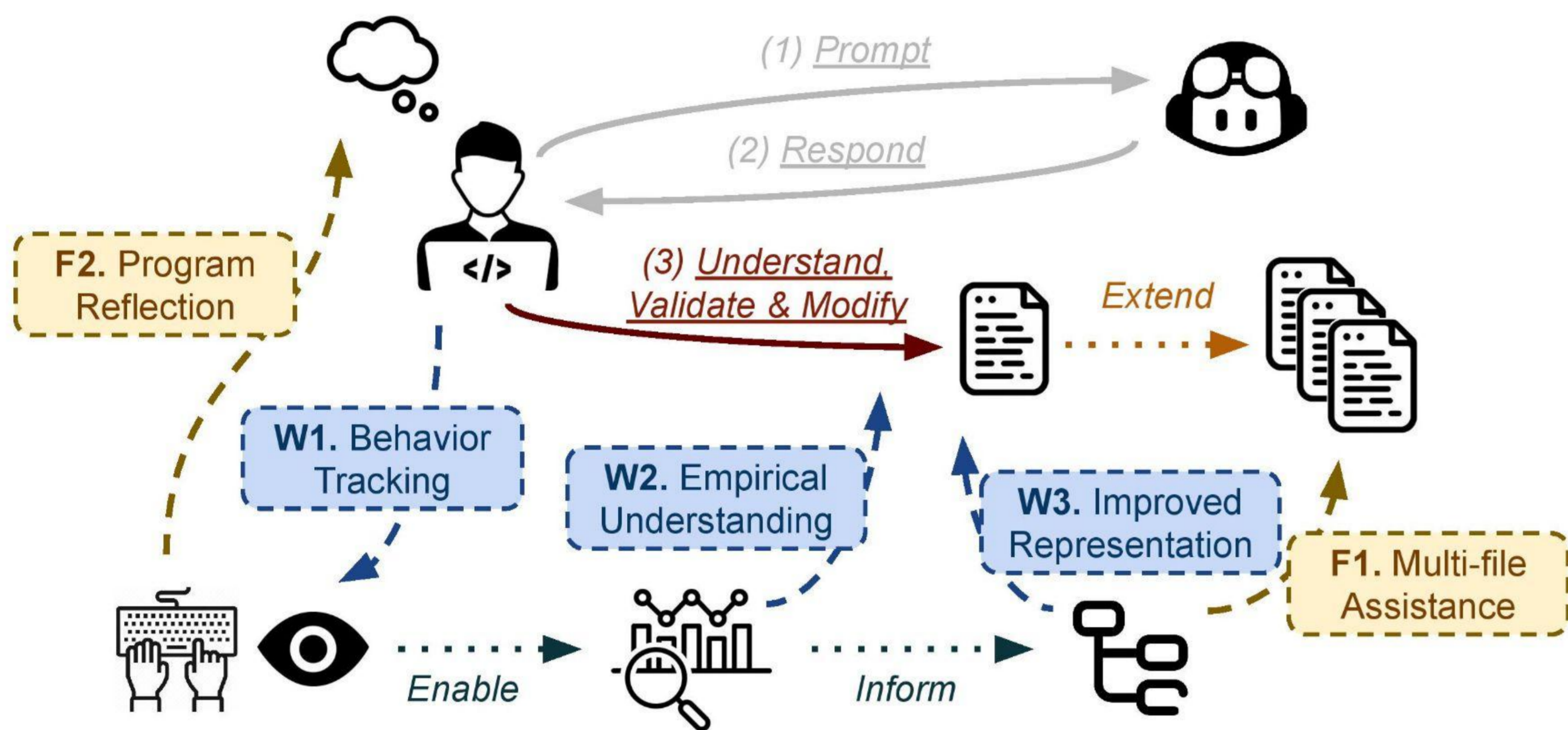
Ningzhi Tang, ntang@nd.edu

University of Notre Dame, Computer Science and Engineering

## Background

Recent advances in large language models (LLMs) are transforming developers' workflows. Developers can now **prompt** the LLMs with their requirements using natural language, and then LLMs **respond** with generated code.

However, the quality of generated code is not guaranteed. Developers must **understand** the code, **validate** its correctness, and **modify** it to integrate it into the existing codebase.



- ⚠️ Uncommon LLM Mistakes
- 🔄 Focus Shift: Details to Structure
- 👁️ Switch Between Code & Prompts
- 💬 Copilot for Comments
- ❓ Can't Distinguish Code
- ✅ Better Validation & Repair
- 👤 More Code Tracing
- 📊 Higher Cognitive Load

## W3. Improved Representation

"Tree Representations Enabled Multi-Level and Syntax-Aware LLM Code Interactions."

```
// Returns the element's index if present in the array; otherwise, returns -1
public static int binarySearch(int[] arr, int element) {
    int left = 0, right = arr.length - 1;
    while (left <= right) {
        int mid = left + (right - left) / 2;
        if (arr[mid] == element) return mid;
        if (arr[mid] < element) left = mid + 1;
        else right = mid - 1;
    }
    return -1;
}
```

Interact With LLM Through

Textual Representation

Tree Representation

Multi-Level Syntax-Aware

## Proposed Future Work

### F1. Multi-file Assistance

Extending **Tree Representation** to Support **Multiple Files**

#### Scenario



**Validate LLM Edits Individually**

Address static and runtime errors

**Syntax-Aware**

**Multi-Level**

**Categorize & Aggregate Edits**

Support validation at different levels of abstraction

### F2. Program Reflection

Enabling **Program Reflection** Through **Behavior Tracking**

IDE Interactions & Eye Tracking

**Low Level Behaviors**

**Observation:** Frequently jump back and forth between multiple files.

Coding Practice Understanding

**High-Level Insights**

**Suggestion:** Use bookmarks or call hierarchy for more efficient navigation.

- 🔧 **Method Gap:** Modeling transition from low-level to high-level insights
- 🌐 **Interface Gap:** Conveying data-driven insights to developers

## Progress to Date

### W1. Behavior Tracking

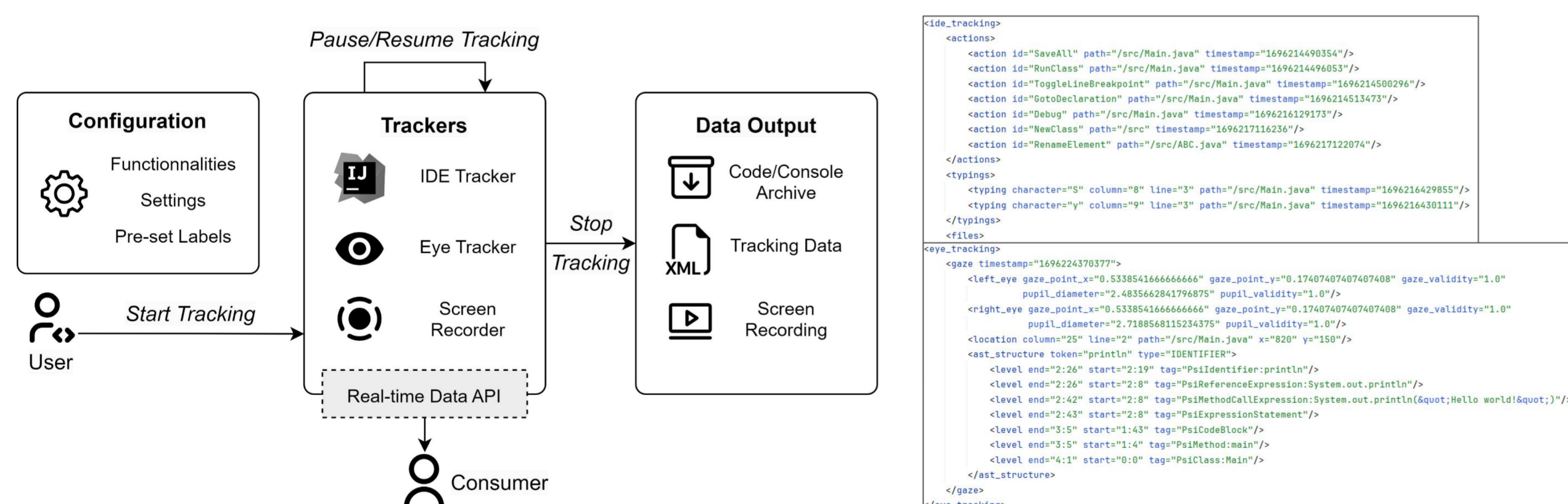
N. Tang\*, J. An\*, M. Chen, A. Bansal, Y. Huang, C. McMillan, T. Li. "CodeGRITS: A Research Toolkit for Developer Behavior and Eye Tracking in IDE." *ICSE-Demo 2024*.

CodeGRITS - **G**aze **R**ecording & **I**DE **T**racking **S**ystem

Supports all **JetBrains IDEs**

**IDE Interactions** & **Eye Movements**

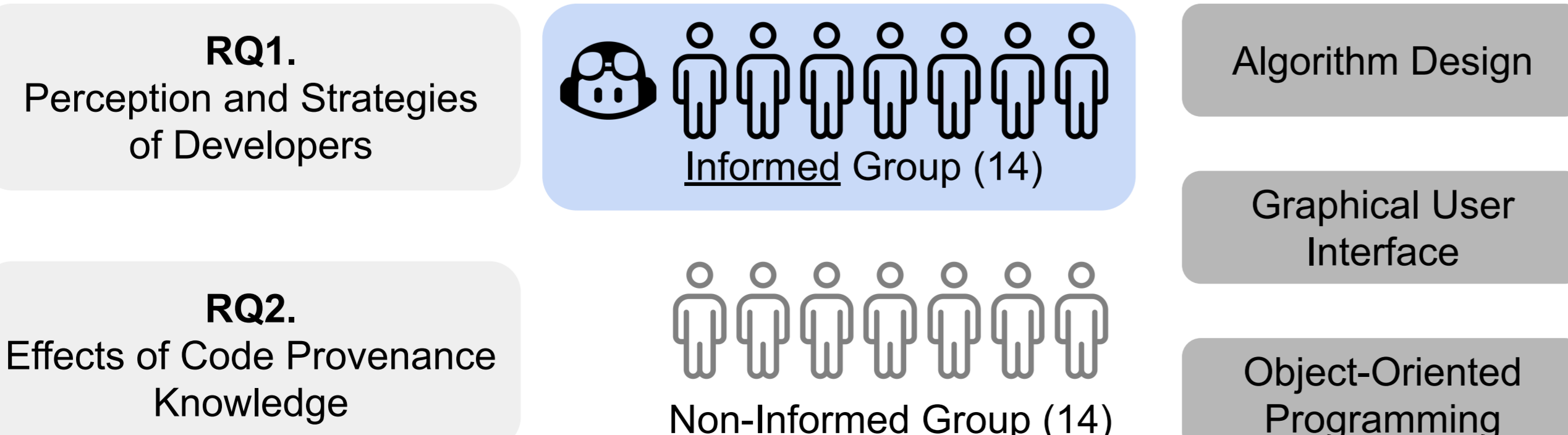
Learn more: [codegrits.github.io/CodeGRITS](https://codegrits.github.io/CodeGRITS)



### W2. Empirical Understanding

N. Tang\*, M. Chen\*, Z. Ning, A. Bansal, Y. Huang, C. McMillan, T. Li. "Developer Behaviors in Validating and Repairing LLM-Generated Code Using IDE and Eye Tracking." *VL/HCC 2024*.

**Validate & Repair LLM-Generated Code**



## Acknowledgement



This project was supported in part by NSF grants CCF-2211428 and CCF-2100035. Any opinions, findings, or recommendations expressed here are those of the authors and do not necessarily reflect the views of the sponsors.