Towards Precise Fault Localization: Spectrum Analysis Powered by Variables and Branches

Hyeongjun Jeon°, Gyeongju Lee, Aditi and Sang-Ki Ko

Computational Intelligence & Data Analytics Laboratory, Department of AI, University of Seoul

UNIVERSITY OF SEOUL

Research Question

"Can statistical fault localization techniques enhance an LLM's ability to repair logical errors?"

Contributions

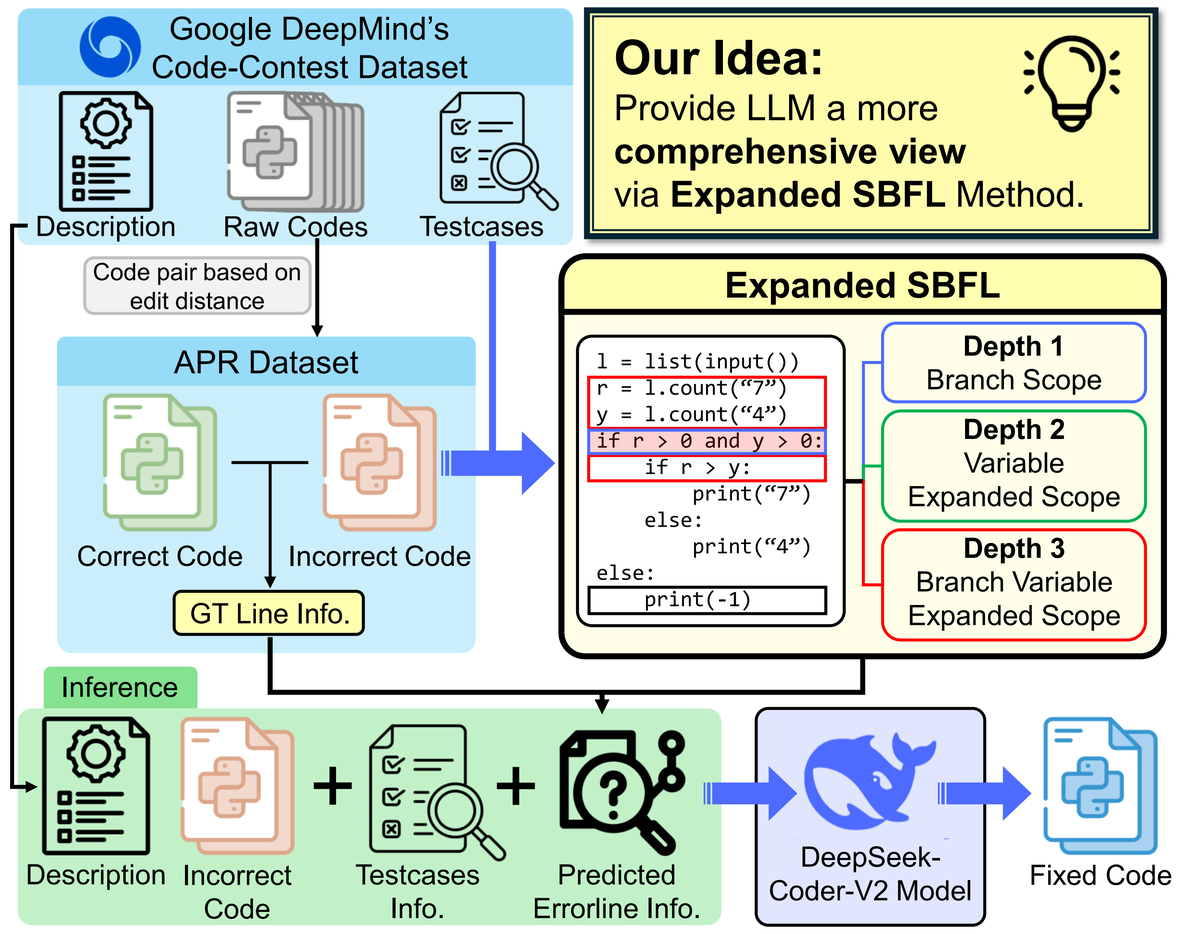
1. New Benchmark Data for Logical Error Repair

- Constructed a dataset for our APR experiments by **pairing** incorrect and correct CodeContest submissions that show **high similarity**, as measured by the **Levenshtein distance**.
- Kept only program pairs whose top-ranked fault was a **conditional statement** (`if/elif`) located **outside any loop**.

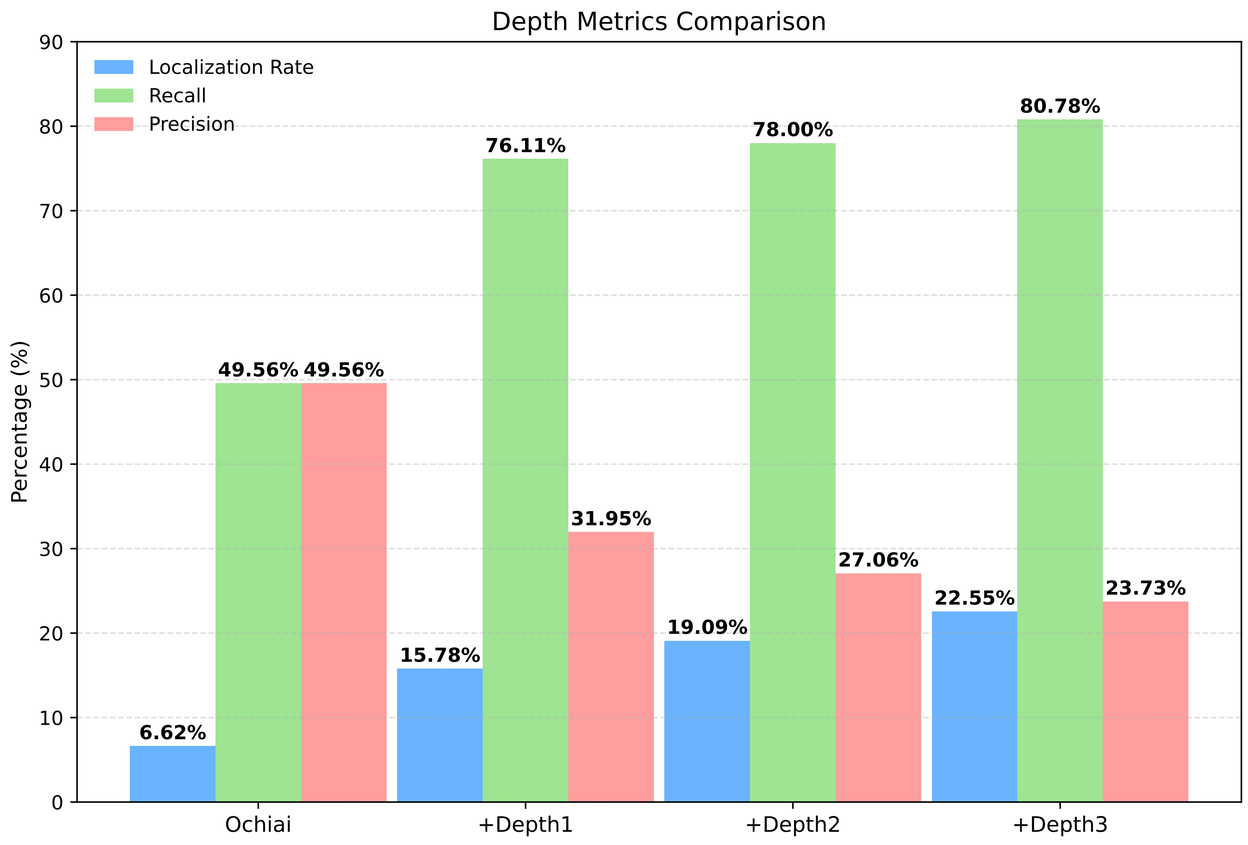
2. Expanded SBFL Method

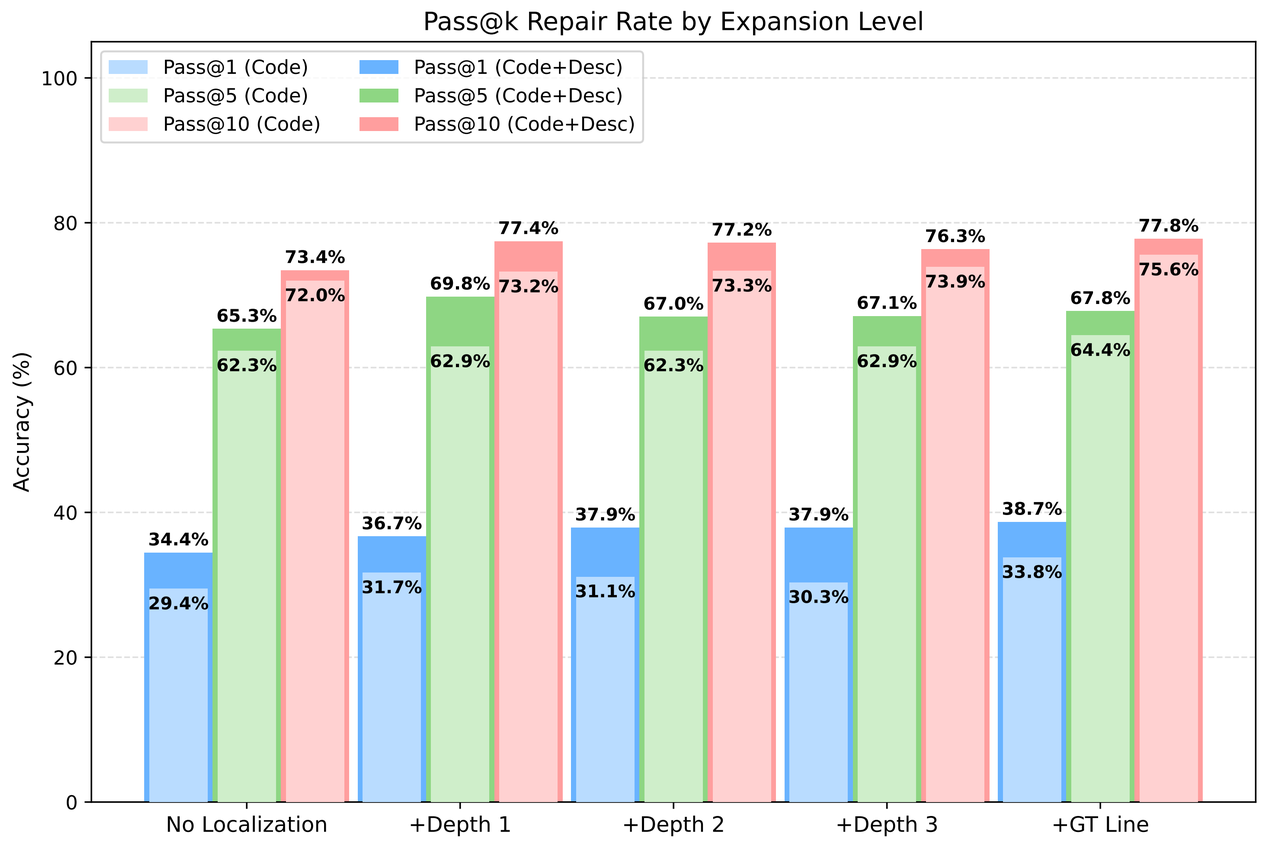
- Proposed **Expanded SBFL**, a novel method that enhances traditional SBFL by expanding the fault context to include related program branches and variable usages.
  1. Depth 1: Includes the entire **control flow branch** (e.g., an if-else block) that contains the initial suspicious line.
  2. Depth 2: Includes all program lines that use any of the **variables** present in the **initial suspicious line**.
  3. Depth 3: The most comprehensive scope, including all lines that use **variables** found in both the **initial line** and the **entire Depth 1 branch**.

Overall Pipeline



Results





$$\text{Localization Rate} = \frac{\sum_{i=1}^{N} |\text{Suspicious}_i|}{\sum_{i=1}^{N} |\text{Total}_i|}, \quad \text{Recall} = \frac{\sum_{i=1}^{N} \mathbf{1}[\text{Fault}_i \cap \text{Suspicious}_i \neq \emptyset]}{\sum_{i=1}^{N} |\text{Fault}_i|}, \quad \text{Precision} = \frac{\sum_{i=1}^{N} \mathbf{1}[\text{Fault}_i \cap \text{Suspicious}_i \neq \emptyset]}{\sum_{i=1}^{N} |\text{Suspicious}_i|}$$

Error Type	Description Only		+ Test Case	
	Base	SBFL+	Base	SBFL+
Output Format Errors	29.5	35.6 (+6.1)	45.3	47.7 (+2.4)
Incorrect Conditional Logic	34.9	35.4 (+0.5)	37.2	39.2 (+2.0)
Loop Errors	35.5	29.0 (-6.5)	37.9	34.5 (-3.4)
Variable Misuse	26.2	28.5 (+2.3)	30.9	33.1 (+2.2)
Errors of Omission	34.5	25.0 (-9.5)	41.0	38.6 (-2.4)
Flawed Calculations	27.8	30.3 (+2.5)	30.3	34.4 (+4.1)
Index Misuse	38.1	31.0 (-7.0)	34.1	38.8 (+4.7)

Conclusions

- Achieved the upper bound of **coverage based APR**: Maximizing bug fix rates at a reasonable cost by fusing **Expanded SBFL** with **rich context** (Description + T.C).
- Boosted performance across most error types: **A strong synergy with T.C.** compensates for SBFL+ weaknesses, though challenges in detecting certain faults remain.
- **Future Work**: Extend the methodology beyond its current scope of specific problems to handle **more complex cases**.

uos/poster cida.uos.ac.kr 113bommy@uos.ac.kr