Fault Localization in LLM-based Multi-Agent Systems

A Spectrum-based Approach using Agent Roles as Spectra

Le Kha Dan Nguyen - nguyen-3@student.tudelft.nl | Supervisors: Burcu Kulahcioglu Ozkan, Annibale Panichella, Zahra Seyedghorban

TU Delft

Introduction

LLM-based Multi-Agent Systems (LLM-MAS): Autonomous programs powered by LLMs where agents plan, use tools, and collaborate to complete tasks.

- LLM-MAS are increasingly deployed in high-stakes settings, yet failure rates remain high, ranging from 41% to 86.7% .
- Failure attribution in LLM-MAS is challenging due to the stochastic behaviors of LLMs and the distributed decision-making process of multi-agent collaboration.

Spectrum-based Fault Localization (SBFL): A popular and lightweight automated diagnosis technique that identifies faulty program components through statistical correlations between system failures and the activity of the different parts of a system.

- SBFL relies on statistical patterns across executions rather than any single execution, which is advantageous for LLM-MAS where stochastic behavior makes individual runs unreliable for fault diagnosis.

Why define the spectrum over agent roles? State-of-the-art multi-agent frameworks e.g. MetaGPT, ChatDev, HyperAgent assign distinct responsibilities to each role, making them well-defined and prospective units for statistical fault correlation.

Research Scope: An empirical evaluation of SBFL as a fault localization technique for agent roles in LLM-MAS, assessing how well this well-understood technique transfers to an agentic setting.

Research Questions

Main Research Question: How effectively can spectrum-based fault localization (SBFL) identify faulty agent roles in LLM-based multi-agent systems (LLM-MAS)?

RQ1.1. How can the core concepts in SBFL be mapped onto agent roles in LLM-MAS execution traces?

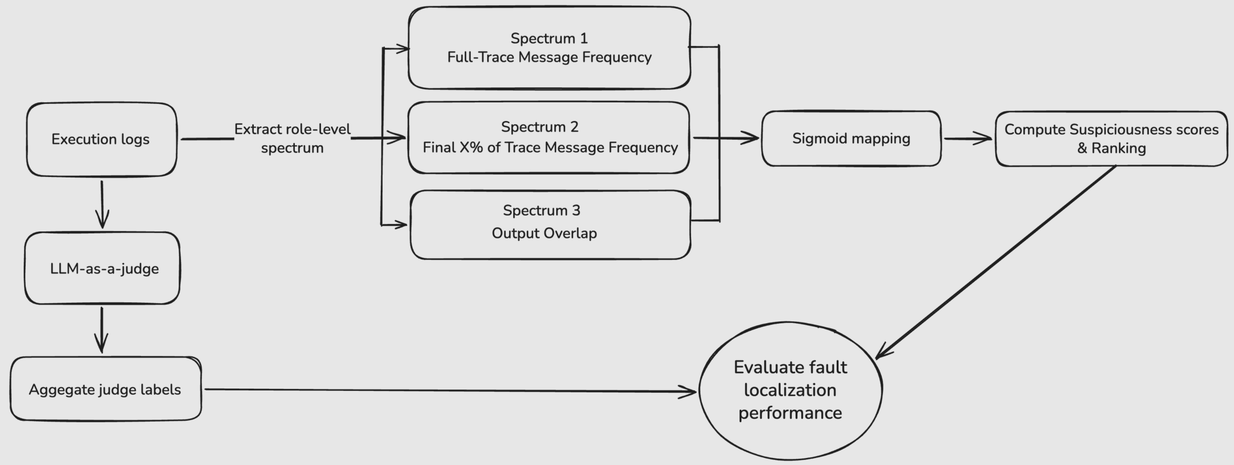
RQ1.2. What constitutes a suitable spectrum for agent roles?

RQ1.3. How accurately does the SBFL suspiciousness ranking identify faulty agent roles, measured by Top-k accuracy against LLM-as-a-judge labels?

Methodology

Methodology Overview: SBFL is built around four core concepts: program components, test cases, test outcomes, and a coverage matrix. By treating agent roles as the analogue of program components, individual executions of LLM-MAS as test cases, and the success or failure of an execution as test outcome, SBFL can be applied to localize faulty agent roles in LLM-MAS.

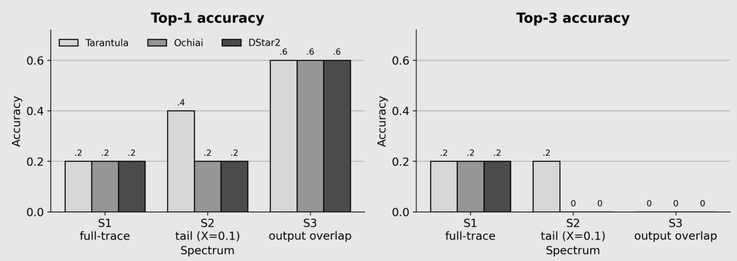
Data Collection: HyperAgent is executed 20 times per task across 5 SWE-bench Verified tasks, producing role-attributed execution logs with automatic pass/fail outcomes.



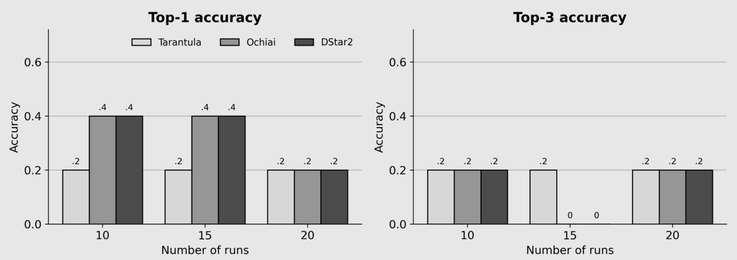
Evaluation Pipelines: The collected execution logs are fed into two parallel pipelines: a SBFL pipeline that extracts role-level participation signals and computes suspiciousness scores and a ground truth pipeline that aggregates LLM-as-a-judge labels into a reference ranking for evaluation.

Results

Fault Localization Accuracy per Spectrum Representation and SBFL Suspiciousness Formula



Effects of Number of Execution Runs Considered on Fault Localization Accuracy



Conclusions & Future Work

Conclusions:

- SBFL can identify faulty agent roles, with semantic output overlap (Spectrum 3) achieving the best Top-1 accuracy of 60%.
- SBFL formula choice has negligible impact on performance.
- Incorporating more execution runs in spectrum analysis does not guarantee improved fault localization performance.

Limitations:

- The evaluation entails only 5 tasks on a single framework.
- LLM-as-a-judge ground truth labels may be inaccurate, limiting the ability to assess SBFL's true effectiveness.

Future Work:

- To scale the experiments to more tasks and LLM-MAS frameworks.
- To replace the LLM-as-a-judge labels with human-annotated labels.
- To experiment with richer spectrum representations.