Cover Letter to the Reviewers

To the Editor,

We thank you and the reviewers for your time in reviewing our manuscript and providing feedback. Your valuable comments and suggestions have led to improvements in the current version of our manuscript. We welcome further feedback if any.

Below, we respond to each of the reviewers' comments.

Sincerely,
Nischal Ashok Kumar
CS PhD Student at UMass Amherst.

Response to Reviewer 1

We thank the reviewer for appreciating that our test case generation method is novel as it works on incorrect codes as well as correct codes which is particularly important in educational settings.

"Give more insights into the iterative prompt engineering. Without much details it is difficult to evaluate what is happening here. Also, is this iterative loop automatic or does someone have to run the code, get results and feed it back to LLMs?"

Response: We propose a fully automatic iterative feedback loop for test case generation which does not require any human intervention thus making it scalable. We add this detail in our introduction and conclusion section. Also, we create a separate section "Iterative Prompt Engineering" in the Appendix to describe engineering details.

"Evaluations are preliminary, and are tested on a simple set of coding challenges. It is still not clear where the system succeeds and where it fails e.g. the paper's explanation on why the system works on "string" and "int array" but not "int" and "boolean" is not convincing."

Response: We thank the reviewer for their concern, although the CSEDM dataset contains simple problems, the bugs demonstrated in the students' code vary across a wide spectrum across different input types like int and string. We further mention that the approach does not work well on "int" and "boolean" as compared to "string" and "int array" because the former categories contain almost double the number of code submissions as compared to the latter categories. For this reason, the former categories represent a wide variety of bugs, hence our code pair selection strategy that selects only three representative code samples per problem might not be effective and we may need to sample more code pairs proportional to the number of examples in that category. We mention this information in the “Results and Discussions” section along with the “Conclusions and Future Work”.

“In Figure 3, while I roughly understand that the items in the middle are where it is not easy to figure out correct vs wrong behavior, and hence many errors occur. It would be great to get a double click into the types of errors that characterize this area”

Response: We thank the author for their comment. We have included more information about the types of common errors for the problem shown in the Analysis section. We have reorganized the Analysis section into two paragraphs.

“There is a big question of how can this generalize to various scenarios and whether that is even possible.”

Response: In this work, we attempt to improve the test case generation ability of LLMs by catering to both fully correct and erroneous student codes. The concept of Langchains or “AI society”, i..e., a collection of LLMs and external systems is becoming popular in improving LLM outputs. Our approach can be generalized by including a diverse code pair selection strategy to improve the pool of generated test cases and can be applied to student codes in various programming languages other than Java.

Response to Reviewer 2

We appreciate the reviewer’s comments that our novel approach has shown success in predicting test cases that accurately measure the student’s ability on the CSEDM dataset.

“It could be an interesting additional (qualitative) analysis to look into how and why some of the intermediate answers have much higher mean errors.”

Response: We have reorganized the Analysis section to include more details on the errors in assignment 494’s problem 46. We also explain how these errors correspond to the measurement of the student’s ability using the predicted test cases.