Milestone 1: Programming Problem Set Curation (2.9 - 2.15)

During this period, we will complete the selection of programming problems required for our research study. Specifically, we will curate 5 representative problems from each difficulty level (easy, medium, and hard) across three major platforms: Codeforces, Luogu, and LeetCode, totaling 15 problems. Additionally, we will select 3 foundational machine learning algorithm problems as a specialized ML test set.

Milestone 2: Automated Evaluation System Development (2.9 - 2.15)

During this period, we will design and implement the automatic evaluation module similar to Online Judge (OJ) platforms. This involves developing the core infrastructure to execute submitted code solutions, handle input/output processing, and assess correctness against predefined test cases. The system will be designed to ensure robust error handling for our experimental framework.

Milestone 3: Test Case Preparation and Accuracy Validation (2.9 - 2.15)

We will prepare comprehensive test cases for each selected problem, including predefined inputs and expected outputs. While no strict quantity requirement is imposed, the test suite will aim for complete coverage of all potential scenarios and edge cases. We will execute each problem's solution code with the prepared inputs and calculate accuracy rates to validate the correctness and reliability of our evaluation framework.

Milestone 4: System Prompt Engineering and Testing (2.23 - 3.1)

During this period, we will engage in iterative dialogue and testing with LLMs to design and refine two critical system prompts. First, we will develop a system prompt that enables the LLM to generate high-quality, structured code explanations and implementation strategies. Second, we will create a system prompt that leverages these structured explanations to produce functional code capable of passing all test cases. Both prompts will be validated through extensive testing to ensure reliability and effectiveness in our experimental pipeline.

Milestone 5: Semantically Equivalent Explanation Generation (3.2 - 3.8)

During this period, we will generate k semantically equivalent explanations (E) for each problem and its corresponding standard solution. These explanations will convey the same algorithmic logic and implementation approach but with varied linguistic expressions. Each explanation E will be independently provided to a fresh GPT instance (without prior context contamination) to generate corresponding code solutions C. This process will enable us to evaluate the consistency and robustness of LLM code generation across semantically equivalent prompts.

Milestone 6: Semantic Similarity Analysis and Documentation (3.9 - 3.15)

During this period, we will conduct pairwise semantic similarity comparisons among all generated explanations. We will compute similarity scores (sim) between each pair of the k explanations and calculate the average pairwise similarity across the entire set to derive a semantic equivalence metric (s). This metric will quantify the degree of semantic consistency among the generated explanations. All similarity scores and the final semantic equivalence metrics will be systematically recorded and preserved for subsequent analysis and reporting.

Milestone 7: Stability Analysis and Data Interpretation (3.16 - 3.22)

During this period, we will conduct a comprehensive stability analysis by integrating code performance scores with semantic equivalence metrics. We will evaluate the correlation between semantic similarity (s) and code correctness rates to assess the robustness and consistency of LLM-generated solutions. A thorough interpretation of the complete dataset will be performed, including statistical analysis, identification of patterns and anomalies, and derivation of insights regarding model stability across varying prompt formulations. Findings will be documented to support our research conclusions.