Appendix: Trail and error

There are various methods that could potentially improve the robustness of LLMs. For example, one approach is to retrain the examined LLMs. However, this method might impair the existing capabilities of the LLM. Moreover, given the enormous number of parameters in LLMs, this approach poses a significant challenge for practical implementation. Alternatively, we attempt to have LLMs perform code refactoring on buggy codes for readability improvement via a direct inference manner. This method aims to retain the essential logic in buggy code while removing unnecessary information before being fed for APR. However, our experiments show that this approach led to a 20.5% reduction in the *R-score* of LLMs, indicating that LLMs struggle to refactor code without adequate training. Additionally, designing reverse rules might improve code readability. However, in practice, the specific perturbation rules of codes are diverse and infinite, while designing each rule manually would consume substantial effort and restrict scalability. Our fine-tuning method, bypassing manual efforts, can effectively overcome the above limitations, allowing LLMs to enhance their generality through supervised learning without tuning their inner parameters.