基于大语言模型参数高效微调的自动化代码审查方法

陆俊逸，于磊，李晓佳，杨立*，左春

The 34th IEEE International Symposium on Software Reliability Engineering (ISSRE 2023), 2023

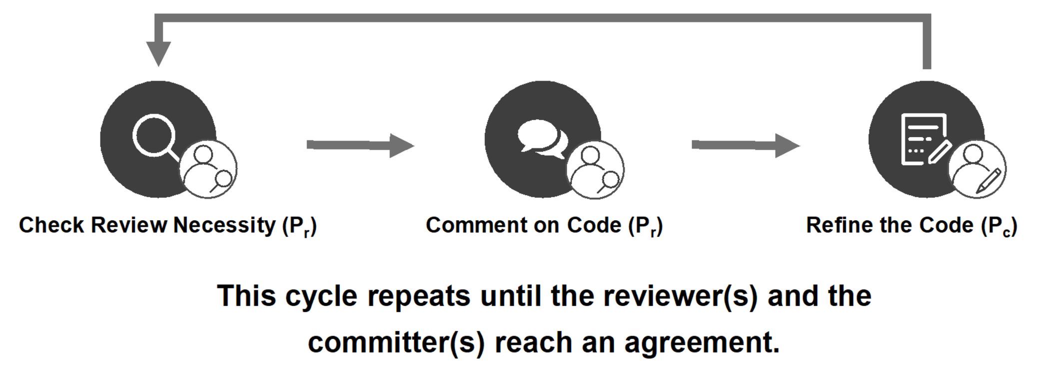
Tel：杨立 yangli2017@iscas.ac.cn 010-62661198

Background

- **Code review** serves as a cornerstone of software engineering, facilitating defect identification, quality improvement, and knowledge sharing.
- Recent advancements in **Large Language Models (LLMs)** have opened up new possibilities for automating code review.

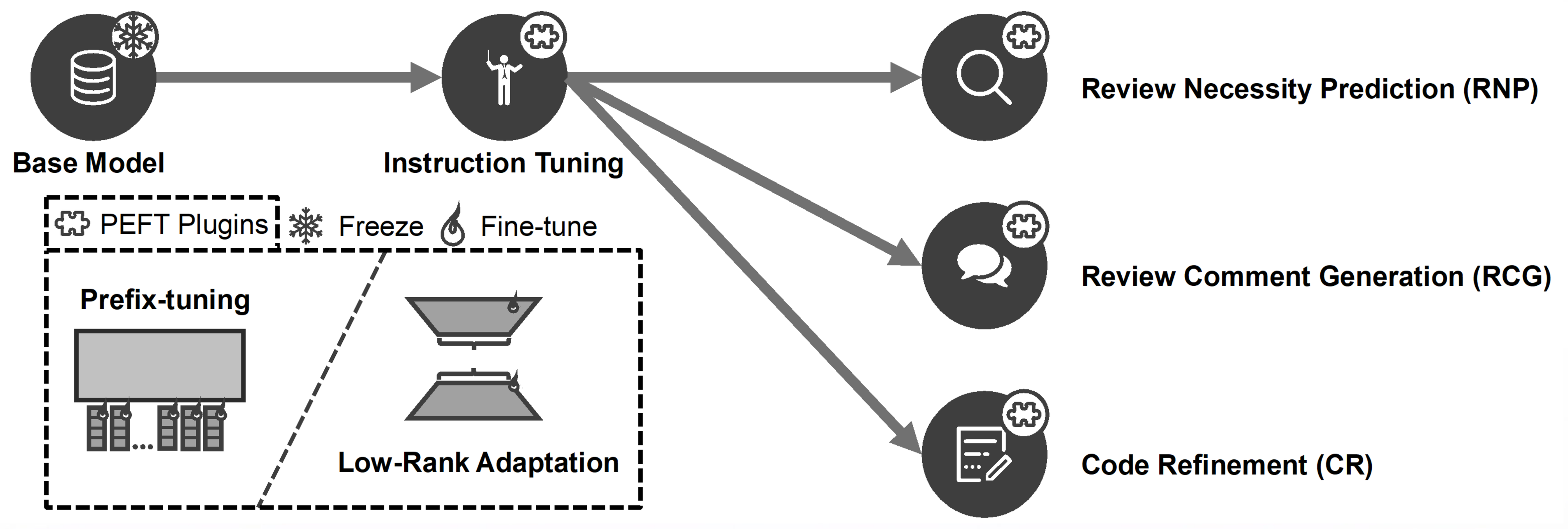
Motivation

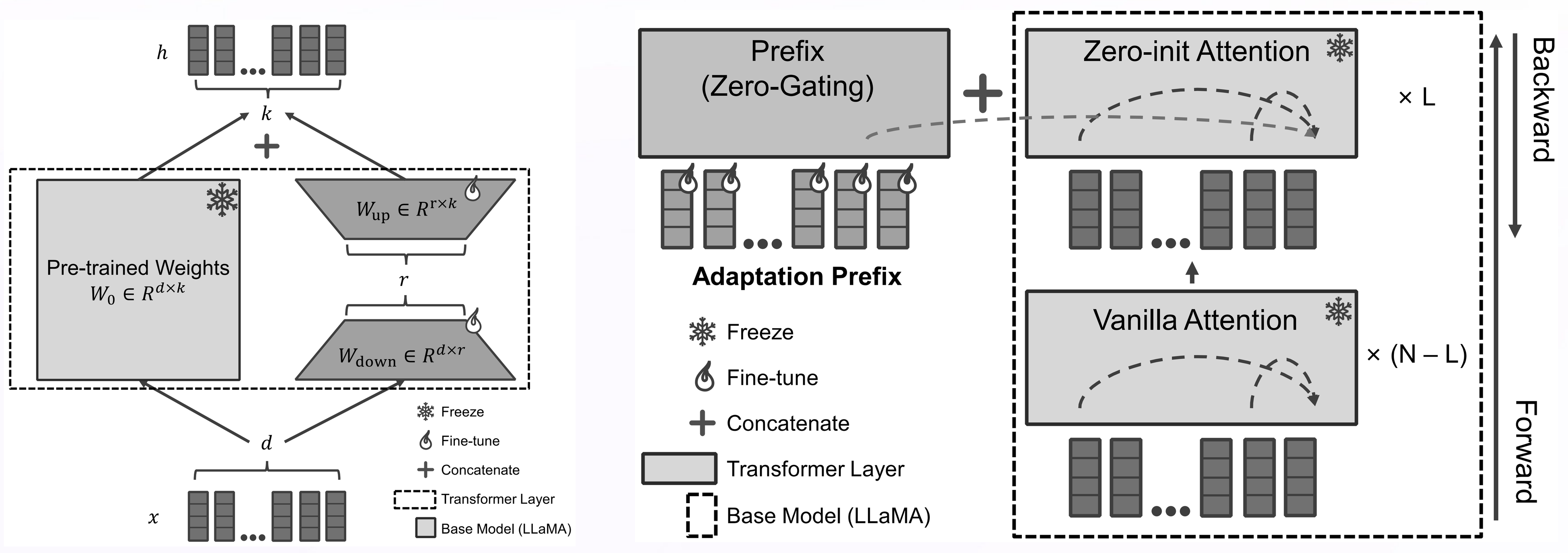
- Computational challenges.
- Improve the efficiency and effectiveness of the code review process.
- While maintaining high quality standards.



Approach: LLaMA-Reviewer

- **Pipeline:** A pipeline process consisting of necessity prediction, review comment generation, and code refinement.
- **Parameter-Efficient Fine-Tuning:** Two types of PEFT methods, including Low-Rank Adaptation and a Zero-init Prefix-Tuning.
- **Data:** Two mainstream datasets, > 100k case on each task.





Evaluation

- **RQ1: Evaluating the Performance of LLaMA-Reviewer**
- LLaMA-Reviewer relatively more excels in generating review comments (NL) and identifies more problems in necessity prediction while maintaining competitive performance in code refinement.
- **RQ2: The Influence of Input Representation**
- LLaMA-Reviewer performs better when the input representation resembles that used during pre-training.
- **RQ3: The Impact of Instruction Tuning**
- Instruction tuning can potentially enhance task performance or the capacity to handle additional natural language information. However, the effectiveness is minor due to the word habit inconsistency among instruction and downstream datasets.
- **RQ4: Influence of Parameter-Efficient Fine-Tuning**
- Among PEFT methods, LoRA is more suitable.