A Prompt Templates

Here are the prompt templates for the Planner LLM (see Figure 1, *left*) to generate change plans; and Reasoner LLM (Figure 1, *right*), to perform intent-aware impact analysis as per the *plan-then-predict* strategy adopted in RIPPLE.

Prompt to Planner LLM for Generating Change Plans

Task. Given an issue description and the initial location that needs to be modified, create a technical summary of the changes needed to fix the issue. Focus on:
- What modifications need to be made
- Where similar changes might be needed (e.g., other parts of the same module, related functions, or code paths affected by the issue)
- Any dependent updates potentially required
- Do not bother about any updates to tests

Input Format.
- Here is a summary of the issue/ticket:
<issue-summary>
- Detailed description of the issue/ticket:
<issue-description>
- Source code at the initial or seed editing location in <file-name>
<seed-code>

Output Format.
<change-plan>
[Provide a high-level summary of what changes may need to potentially be made to fix the issue, discussing key impact points and potential ripple effects.]
</change-plan>
If there is insufficient information to make a definitive determination, make required assumptions while clearly stating them. Respond only with <change_plan> tags as described above. Do not include any other text in your response.

Prompt to Reasoner LLM for Intent-Aware Change Impact Analysis

For the given issue/ticket, based on the following change plan (a first draft), determine which of the methods provided in the repository structure will be impacted, i.e., will need changes to be made in.

Input Format.
- Here is a summary of the issue/ticket:
<issue-summary>
- A first draft for the change plan:
<change-plan>
- Repository structure with method summaries
<repository-structure>

Task.
- Identify impacted methods from the repository structure based on their role and dependencies.
- Consider how class structure affects impact propogation.
- Justify the impact decision for each method. If impact is unclear, state assumptions.

Output Format.
<impacted-methods>
class_name1,method_name1
class_name1,method_name2
class_name2,method_name3
...
</impacted-methods>
Your assessment should be based solely on the provided Java repository structure. Note that you may need to make educated guesses about method interactions, and the focus is on precise identification.

Figure 1: Prompts to LLMs in RIPPLE for: (*left*) generating change plans, and (*right*) performing intent-aware impact analysis.

B Sensitivity Analysis

Table 1: Sensitivity analysis based on:

#-Hops	Evaluation Metrics (in %)			
	Hit@k	*Prec.*	*Recall*	*F1*
0	81.0	8.8	62.7	12.6
1	**86.0**	7.6	64.7	11.1
2	86.0	7.3	65.9	10.7
3	86.0	7.2	66.0	10.5
4	86.0	7.3	66.5	10.7

(a) number of hops in indirect dependence coupling

N	Evaluation Metrics (in %)			
	Hit@k	*Prec.*	*Recall*	*F1*
50	32.0	5.0	17.1	5.2
75	32.0	5.0	17.1	5.2
100	32.0	4.9	17.1	5.2
125	32.0	5.0	17.1	5.2
150	32.0	5.0	17.1	5.2

(b) previous N commits in evolutionary coupling

B.1 Number of Hops in Indirect Dependence Coupling

As mentioned in Section 5.2.1 in the paper, by default, we limit the dependence depth L in the recall-focused impact set expansion phase in RIPPLE (*i.e.*, $\mathcal{I}_H \rightarrow \mathcal{I}_D$) to 1 hop. In Table 1a, we present a sensitivity analysis based on the number of hops in indirect dependence coupling. We can see that as L increases, the size of the impact sets captured by indirect dependence coupling increases–resulting in a decrease in precision. For this reason, we opt for only one hop in indirect dependence coupling.

B.2 Number of Previous Commits in Evolutionary Coupling

As mentioned in Section 3.2.2 in the paper, we truncate the commit history by considering only the most recent 100 relevant commits (*i.e.*, those where the initial seed edit location was modified). This conforms with the principle of locality that the files that have been changed recently are likely to be changed in the near future. In Table 1b, we present a sensitivity analysis based on the number of relevant previous commits. We can see that as N increases, beyond $N = 100$, I_D results in similar performance. For this reason, as a default setting, we chose $N = 100$ in the *recall-focused impact set expansion* phase in RIPPLE.