4.4 General RL

The General RL stage aims to broadly enhance the models' capabilities and stability across diverse scenarios. To facilitate this, we have established a sophisticated **reward system** covering **over 20 distinct tasks**, each with customized scoring criteria. These tasks specifically target enhancements in the following core capabilities:

- **Instruction Following**: This capability ensures that models accurately interpret and follow user instructions, including requirements related to content, format, length, and the use of structured output, delivering responses that align with user expectations.
- **Format Following**: In addition to explicit instructions, we expect the model to adhere to specific formatting conventions. For instance, it should respond appropriately to the `/think` and `/no_think` flags by switching between thinking and non-thinking modes, and consistently use designated tokens (e.g., `<think>` and `</think>`) to separate the thinking and response parts in the final output.
- **Preference Alignment**: For open-ended queries, preference alignment focuses on improving the model's helpfulness, engagement, and style, ultimately delivering a more natural and satisfying user experience.
- **Agent Ability**: This involves training the model to correctly invoke tools via designated interfaces. During the RL rollout, the model is allowed to perform complete multi-turn interaction cycles with real environment execution feedback, thereby improving its performance and stability in long-horizon decision-making tasks.
- **Abilities for Specialized Scenarios**: In more specialized scenarios, we design tasks tailored to the specific context. For example, in Retrieval-Augmented Generation (RAG) tasks, we incorporate reward signals to guide the model toward generating accurate and contextually appropriate responses, thereby minimizing the risk of hallucination.

To provide feedback for the aforementioned tasks, we utilized three distinct types of rewards:

(1) **Rule-based Reward**: The rule-based reward has been widely used in the reasoning RL stage, and is also useful for general tasks such as instruction following (Lambert et al., 2024) and format adherence. Well-designed rule-based rewards can assess the correctness of model outputs with high precision, preventing issues like reward hacking.

(2) **Model-based Reward with Reference Answer**: In this approach, we provide a reference answer for each query and prompt Qwen2.5-72B-Instruct to score the model's response based on this reference. This method allows for more flexible handling of diverse tasks without requiring strict formatting, avoiding false negatives that can occur with purely rule-based rewards.

(3) **Model-based Reward without Reference Answer**: Leveraging human preference data, we train a reward model to assign scalar scores to model responses. This approach, which does not depend on a reference answer, can handle a broader range of queries while effectively enhancing the model's engagement and helpfulness.

4.5 Strong-to-Weak Distillation

The Strong-to-Weak Distillation pipeline is specifically designed to optimize lightweight models, encompassing 5 dense models (Qwen3-0.6B, 1.7B, 4B, 8B, and 14B) and one MoE model (Qwen3-30B-A3B). This approach enhances model performance while effectively imparting robust mode-switching capabilities. The distillation process is divided into two primary phases:

(1) **Off-policy Distillation**: At this initial phase, we combine the outputs of teacher models generated with both `/think` and `/no_think` modes for response distillation. This helps lightweight student models develop basic reasoning skills and the ability to switch between different modes of thinking, laying a solid foundation for the next on-policy training phase.

(2) **On-policy Distillation**: In this phase, the student model generates on-policy sequences for fine-tuning. Specifically, prompts are sampled, and the student model produces responses in either `/think` or `/no_think` mode. The student model is then fine-tuned by aligning its logits with those of a teacher model (Qwen3-32B or Qwen3-235B-A22B) to minimize the KL divergence.

4.6 Post-training Evaluation

To comprehensively evaluate the quality of instruction-tuned models, we adopted automatic benchmarks to assess model performance under both thinking and non-thinking modes. These benchmarks are