

Appendix

Group Relative Policy Optimization (GRPO)

We utilize the Group Relative Policy Optimization (GRPO) as the RL algorithm. For each question $x \sim \mathcal{D}$, the behavior policy $\pi_{\theta_{\text{old}}}$ generates a set of G candidate completions $\tau = \{y_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|x)$, with each response receiving a scalar reward r_i . The training objective is to optimize the policy π_{θ} based on reference policy $\pi_{\theta_{\text{ref}}}$:

$$\mathcal{J}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \{y_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|x)} \frac{1}{G} \sum_{i=1}^G [\min(\frac{\pi_{\theta}(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)} \hat{A}_i, \text{clip}(\frac{\pi_{\theta}(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)}, 1 - \epsilon, 1 + \epsilon) \hat{A}_i) - \beta \mathbb{D}_{\text{KL}}(\pi_{\theta} || \pi_{\theta_{\text{ref}}})]$$
(6)

where the group-normalized advantage \hat{A}_i of the i -th rollout in current group is defined as:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G)}$$

An overview of the GRPO algorithm is illustrated in Figure 4. In this formulation, ϵ denotes the clipping ratio, a hyperparameter that controls the allowable deviation between the updated and reference policies. The `clip` function restricts the importance weight r_i within the range $[1 - \epsilon, 1 + \epsilon]$, which enhances training stability and reduces the risk of policy collapse. The parameter β represents the Kullback–Leibler (KL) loss coefficient (Hall 1987), which governs the strength of the KL divergence penalty included in the objective function. This penalty term helps constrain the policy updates, ensuring that the learned policy remains sufficiently close to the original reference policy and thereby improving overall training stability.

Experiment Details

Datasets

To evaluate the performance of PilotRL, we conduct experiments using six datasets for agent tasks. Specifically, four datasets are used for training and in-domain (ID) performance evaluation, while the remaining two are reserved for out-of-domain (OOD) assessment, as shown in Table 1.

- **ALFWorld** (Shridhar et al. 2021): ALFWorld is a home-oriented environment built upon TextWorld, where agents are required to navigate through rooms and apply common sense reasoning to perform various tasks. It mirrors the embodied settings found in the ALFRED dataset (Shridhar et al. 2020), and offers human-annotated ideal trajectories for use in imitation learning.
- **IQA** (Gordon et al. 2018): The Interactive QA dataset is a question answering task in which an agent need to engage with a dynamic visual environment to find answers. Here we utilize the text version from Jia et al. (2024).
- **TextCraft** (Prasad et al. 2024): It is a text-only environment for crafting Minecraft items that resembles cooking

recipes with steps of varying complexity. This dataset exhibits an inherently decomposable structure, providing a more suitable environment for our proposed paradigm.

- **Wordle** (Abdulhai et al. 2023): It is a word-guessing game designed to assess agents’ reasoning capabilities at the letter level, where the agents attempt to identify a target word selected from a predefined vocabulary consisting of five-letter words. In order to successfully identify the target word with minimum trials within the limited number of allowed attempts, it is crucial for the model to employ efficient global planning.
- **MAZE** (Abdulhai et al. 2023): MAZE is also a word-based puzzle game in which agents, serving as players, are aware of their current position, the location of the goal, and the presence of walls in the four cardinal directions (up, down, left, and right).
- **BabyAI** (Chevalier-Boisvert et al. 2019): The BabyAI dataset evaluates agent performance in embodied navigation and interaction scenarios. It features a simulated grid-world environment containing 40 instruction-following tasks, where agents are required to understand commands and interact with objects accordingly.

We collected the training and evaluation data from Song et al. (2024) and Xi et al. (2024). For ALFWorld and IQA, we utilize the datasets as provided in Song et al. (2024), while for TextCraft, Wordle, MAZE, and BabyAI, we adopt the versions from Xi et al. (2024). The reference trajectories included in these original data sources are used exclusively for supervised fine-tuning (SFT) of the baselines. During both the reinforcement learning (RL) training and evaluation phases, we only make use of the task instructions and their corresponding final answers.

Baselines

In this section, we provide a comprehensive overview of the various methods that serve as baselines in our comparison.

- **Close-Sourced Models:** Closed-source models are considered to represent the current state-of-the-art in LLM capabilities and are regarded as the most competitive baseline methods. We have selected GPT-4o and GPT-4o-mini (Hurst et al. 2024) to assess the upper bound of the model performance on agent tasks.
- **Open-Sourced Agent-Specific Models:** These models refer to models that were trained specifically on agent-task datasets. We have selected Agent-FLAN-7B (Chen et al. 2024), LLaMA-xLAM-2-8B-fc-r (Zhang et al. 2024a) and DeepResearcher-7B (Zheng et al. 2025) to represent the open-sourced agent-specific models for comparison to assess PilotRL’s relative advantages. Specifically, the backbone model of DeepResearcher-7B is Qwen2.5-7B-Instruct (Yang et al. 2024), which facilitates a more direct comparison with Qwen2.5-7B-Instruct + PilotRL.
- **Naive Response:** It refers to the case where the model directly generates responses without any training (e.g., SFT, RL, etc.) or prompting (e.g., ReAct) strategies.

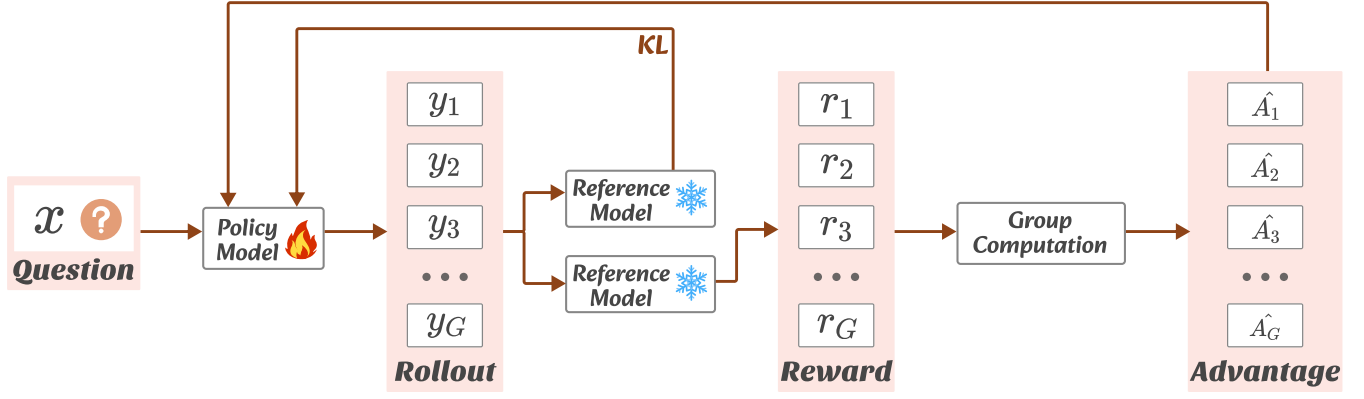


Figure 4: An illustration for the Group Relative Policy Optimization (GRPO) pipeline.

- **ReAct** (Yao et al. 2023): It is the prompting strategy that integrates single-step reasoning with the execution of the current action, which is a common agent paradigm.
- **MPO** (Xiong et al. 2025): The Meta Plan Optimization (MPO) framework improves the agent’s planning capabilities by integrating explicit guidance into the decision-making process. As an external plug-and-play planner, MPO provides the model with high-level meta-plans that serve as structured guidance during task execution. One key distinction between MPO and PilotRL lies in the integration and training of the planner and executor components. In our approach, both the planner and executor reside within the same model and are trained jointly. In contrast, MPO maintains separate models for planning and execution, where only the planner is trained while the executor’s parameters remain frozen, leading to limited coordination between the two components.
- **Supervised Fine-Tuning (SFT)**: This training strategy is widely adopted in a series of studies (Chen et al. 2024; Song et al. 2024; Xi et al. 2024; Zeng et al. 2024; Zhang et al. 2024b; Fu et al. 2025). However, existing studies have shown that compared to RL, SFT generally exhibits weaker generalization capabilities on new tasks—particularly when the training data consists of multi-step trajectories for problem-solving (Shao et al. 2024; Team et al. 2025). This is because such trajectories may contain redundant or suboptimal paths to task completion. Moreover, SFT tends to bias the model toward previously seen execution paths, limiting its ability to adapt or generalize to novel scenarios through compositional or analogical reasoning. During SFT, we use the same datasets with PilotRL. In addition, we incorporate the original agent-environment interaction trajectories into training, a setting that differs from **Vanilla RL** and our **PilotRL**. Furthermore, we generate global plans for guiding task completion using DeepSeek-V3, and feed both the interaction trajectories and the corresponding global plans into the model during training. This setup allows us to compare PilotRL over existing baselines under a more fair and controlled experimental condition.

- **Vanilla RL**: We also conduct training with the naive reinforcement learning process utilizing the Group Relative Policy Optimization (GRPO) (Shao et al. 2024) algorithm. Here we employ only the format and end-to-end (E2E) performance as the reward metrics. This baseline is for validating the effectiveness of adaptive global planning.

Ablation Study Details

We report the original performance scores of the models on each benchmark during training stage and training sequence ablation, as well as reward design ablation, as depicted in Table 6, Table 7 and Table 8.

Declaration for Figure 3 It is worth noting that when analyzing the evolution of reward scores for the global planner, the executor, and the end-to-end (E2E) performance using LLaMA3.1-8B-Instruct + PilotRL, we normalized all reward scores to the range $[0, 1]$ for visualization and comparison purposes. The reward metrics include:

- **Global Planner**: This reward function (Equation (5)) is introduced starting from Stage 2, and operates during Stage 2 (epoch 2 & 3). In Stage 3, we only evaluate and record this metric without using it for model updates.
- **Executor**: This reward (Equation (3)) is used as the training objective solely in Stage 1. In subsequent stages, we continue to log its value for analysis, but it no longer influences model updates.
- **End-to-End (E2E) Performance**: The reward based on end-to-end performance (Equation (4)) is evaluated throughout the entire training process and serves as a consistent metric for assessing overall system behavior.

Environment and Hardware Configurations

The experiment utilizes the following core libraries and their respective versions: torch=2.5.1, CUDA_version=12.4, ray=2.40.0, vllm=0.7.3, verl=0.2.0.post2, transformrs=4.49.0, datasets=3.3.2, tqdm=4.40.0, flash-attn=2.5.8, pyarrow=19.0.1, tensordict=0.5.0. Experiments are conducted using 32 NVIDIA H20 GPUs with 96GB memory.

Order	Backbone Model	ALFWorld	IQA	TextCraft	Wordle	BabyAI	MAZE	Avg.
		In-Domain (ID)				Out-of-Domain (OOD)		
Standard Pipeline								
1 → 2 → 3 (ours)	Qwen2.5-7B-Instruct	70.80	67.84	75.37	77.69	61.56	57.93	68.53
	LLaMA3.1-8B-Instruct	78.53	72.78	64.76	79.61	68.24	58.68	70.43
	Qwen3-8B	72.51	69.06	71.48	83.65	65.28	56.62	69.77
Necessity of Progressive Training								
1 & 2 & 3	Qwen2.5-7B-Instruct	68.29	65.43	72.91	75.82	57.98	54.37	65.80
	LLaMA3.1-8B-Instruct	75.56	70.42	63.03	74.51	63.74	56.00	67.21
	Qwen3-8B	70.89	71.30	69.68	81.84	63.19	55.81	68.79
Effectiveness of Stage 1 (Instruction Adherence)								
2 → 3	Qwen2.5-7B-Instruct	66.37	63.85	72.16	74.93	60.05	52.54	64.98
	LLaMA3.1-8B-Instruct	73.86	70.19	63.75	72.66	64.37	54.93	66.63
	Qwen3-8B	70.97	69.63	70.12	81.35	63.96	54.10	68.36
Effectiveness of Stage 2 (Global Planner Cultivation)								
1 → 3	Qwen2.5-7B-Instruct	66.72	66.38	71.74	76.56	58.85	53.48	65.62
	LLaMA3.1-8B-Instruct	73.04	72.43	61.59	70.47	66.32	53.26	66.19
	Qwen3-8B	70.56	68.36	69.04	80.98	64.47	53.95	67.89
Effectiveness of Stage 3 (Dual-Process Collaboration)								
1 → 2	Qwen2.5-7B-Instruct	67.49	65.82	75.65	73.34	60.78	53.17	66.04
	LLaMA3.1-8B-Instruct	75.40	71.55	62.88	75.67	65.19	56.92	67.94
	Qwen3-8B	72.18	72.61	70.59	83.27	64.73	53.28	69.44
Sequential Order of Stages								
2 → 1 → 3	Qwen2.5-7B-Instruct	70.12	66.08	73.98	77.85	59.63	55.67	67.22
	LLaMA3.1-8B-Instruct	77.25	73.15	64.02	77.63	65.98	58.14	69.36
	Qwen3-8B	72.94	73.86	68.55	78.02	65.07	54.80	68.87

Table 6: **Original scores for each benchmark of the ablation study on multiple training stages and sequential order.** It is the detailed version of Table 3. “Order” is the sequential order of Stage 1, 2, and 3 during training. Specifically, “1 & 2 & 3” refers to a joint training configuration in which reward functions from all three stages are merged and optimized concurrently, where the target model generates global plans independently throughout the entire training process. The best and second best scores of each model are in **bold** and underlined.

Backbone Model	Paradigm	ALFWorld	IQA	TextCraft	Wordle	BabyAI	MAZE
		<i>In-Domain (ID)</i>				<i>Out-of-Domain (OOD)</i>	
Qwen2.5-7B-Instruct	ReAct	52.15	37.57	34.46	40.43	44.08	37.52
	AdaPlan	59.72 (↑ 14.52%)	43.68 (↑ 16.26%)	45.54 (↑ 32.15%)	53.23 (↑ 31.66%)	47.90 (↑ 8.67%)	42.05 (↑ 12.07%)
LLaMA3.1-8B-Instruct	ReAct	38.48	42.94	45.83	38.56	47.36	36.92
	AdaPlan	44.19 (↑ 14.84%)	48.02 (↑ 11.83%)	46.67 (↑ 1.83%)	50.78 (↑ 31.69%)	54.46 (↑ 14.99%)	39.94 (↑ 8.18%)
Qwen3-8B	ReAct	62.56	50.58	44.62	41.60	54.35	42.68
	AdaPlan	63.34 (↑ 1.25%)	53.82 (↑ 6.41%)	44.98 (↑ 0.81%)	52.61 (↑ 26.47%)	55.73 (↑ 2.54%)	47.24 (↑ 10.68%)

Table 7: **Original scores for each benchmark of the agent paradigm analysis.** It is the detailed version of Table 4. The best scores of each model are in **bold**. It shows that *AdaPlan* consistently outperforms *ReAct* on both in-domain and out-of-domain agent tasks across all models, demonstrating performance gains of 18.64%, 13.58%, 7.19% on Qwen2.5-7B-Instruct, LLaMA3.1-8B-Instruct, and Qwen3-8B, respectively.

Prompts

Here we present the prompts used throughout our pipeline in PilotRL . Only the English version is presented due to LaTeX compilation issues with non-English languages.

Prompt: Global Plan Generation - ALFWorld

Backbone Model	Architecture	ALFWorld	IQA	TextCraft	Wordle	BabyAI	MAZE
		<i>In-Domain (ID)</i>				<i>Out-of-Domain (OOD)</i>	
Qwen2.5-7B-Instruct	Isolated	68.85	64.18	72.60	70.14	58.29	52.07
	Unified	70.80 (↑ 2.83%)	67.84 (↑ 5.70%)	75.37 (↑ 3.82%)	77.69 (↑ 10.76%)	61.56 (↑ 5.61%)	57.93 (↑ 11.25%)
LLaMA3.1-8B-Instruct	Isolated	71.87	70.83	60.96	71.05	62.71	55.64
	Unified	78.53 (↑ 9.27%)	72.78 (↑ 2.75%)	64.76 (↑ 6.23%)	79.61 (↑ 12.05%)	68.24 (↑ 8.82%)	58.68 (↑ 5.46%)
Qwen3-8B	Isolated	71.74	67.71	68.96	82.23	60.55	51.49
	Unified	72.51 (↑ 1.07%)	69.06 (↑ 1.99%)	71.48 (↑ 3.65%)	83.65 (↑ 1.73%)	65.28 (↑ 7.81%)	56.62 (↑ 9.96%)

Table 8: **Original scores for each benchmark of the planner-executor architecture analysis.** It is the detailed version of Table 5. The best scores of each model are in **bold**. It shows that the *unified architecture* consistently outperforms *isolated architecture* on both in-domain and out-of-domain agent tasks across all models, with measured improvements of 6.48%, 7.51%, 3.96% on Qwen2.5-7B-Instruct, LLaMA3.1-8B-Instruct, and Qwen3-8B, respectively.

Based on the task description, the previous global plan, and accumulated observation of agent interactions with the environment, please generate all possible step-by-step global plans, which serve as high-level, natural guidance to assist in planning. Maintain the plan for all steps preceding the execution step index, while selectively modifying the plan for steps following the execution step index.

For house holding task, the action list you can take:

1. go to recep
2. task obj from recep
3. put obj in/on recep
4. open recep
5. close recep
6. toggle obj recep
7. clean obj with recep
8. heat obj with recep
9. cool obj with rece

where obj and recep correspond to objects and receptacles.

Task
{task}

Previous Global Plan
{global_plan} [optional]

Execution Step Index
{execution_step_index}

Accumulated Observation
{observation} [optional]

Output Format:

```
```json
[
 "
 Step 1: ...
 Step 2: ...
 ...
 ", ...]
```
```

Prompt: Global Plan Generation - IQA

Based on the task description, the previous global plan, and accumulated observation of agent interactions with the environment, please generate all possible step-by-step global plans, which serve as high-level, natural guidance to assist in planning. Maintain the plan for all steps preceding the execution step index, while selectively modifying the plan for steps following the execution step index.

For interactive QA task, the action list you can take:

1. move ahead
2. turn left
3. turn right
4. open obj
5. answer [True]/[False]

where obj correspond to objects.

Task
{task}

Previous Global Plan
{global_plan} [optional]

Execution Step Index
{execution_step_index}

Accumulated Observation
{observation} [optional]

Output Format:

```
```json
[
 "
 Step 1: ...
 Step 2: ...
 ...
 ", ...]
 , ...]
 , ...]
```

**Prompt: Global Plan Generation - TextCraft**

You are given a few useful crafting recipes to craft items in Minecraft. Craft command can be understood as follows: craft [target] using [ingredients], where target is item/object generated by the craft command as output and ingredient are the inputs. You are given an agent that can “craft” or “fetch” objects. You can take the help of crafting commands below to create new objects. Based on the task description, the previous global plan, and accumulated observation of agent interactions with the environment, please generate all possible step-by-step global plans, which serve as high-level, natural guidance to assist in planning. Maintain the plan for all steps preceding the execution step index, while selectively modifying the plan for steps following the execution step index. Each global plan can use at most ONE of the provided crafting commands.

**# Task**

{task}

**# Previous Global Plan**

{global\_plan} [optional]

**# Execution Step Index**

{execution\_step\_index}

**# Accumulated Observation**

{observation} [optional]

---

**Output Format:**

```
```json
[
  "
  Step 1: ...
  Step 2: ...
  ...
  ", ...]
  , ...]
  , ...]
```

Prompt: Global Plan Generation - Wordle

You are an expert wordle player. Based on the task description, the previous global plan, and accumulated observation of agent interactions with the environment, please generate all possible step-by-step global plans for the wordle task, which serve as high-level, natural guidance to assist in planning. Maintain the plan for all steps preceding the execution step index, while selectively modifying the plan for steps following the execution step index. Your objective is to guess a hidden 5 letter word. You have 6 attempts to guess it correctly and you should try to guess it in as few attempts as possible. When guessing the word, you should format your word as a space separated sequence of letters, like “s h i r e” for example. After guessing the word, you will receive feedback from the game environment in the form of a sequence of 5 space separated letters like “b y g g b”, where each letter indicates some information about the hidden word. The environment will return one of three letters - “b”, “g”, or “y” – for each letter in the word you guessed. Here is the meaning of each letter:

- “b”: If the environment returns a “b”, it means that the letter at that position in your guessed word is not in the hidden word.
- “y”: If the environment returns a “y”, it means that the letter at that position in your guessed word is in the hidden word but is not in the correct position.
- “g”: If the environment returns a “g”, it means that the letter at that position in your guessed word is in the hidden word and is in the correct position.

Task

{task}

Previous Global Plan

{global_plan} [optional]

Execution Step Index

{execution_step_index}

Accumulated Observation

{observation} [optional]

Output Format:

```
```json
[
 "
 Step 1: ...
 Step 2: ...
```

```
...
"/, ...]
,,,
```

### Prompt: Global Plan Generation - BabyAI

You are an exploration master that wants to finish every goal you are given. You are placed in a room and you need to accomplish the given goal with actions. Based on the task description, the previous global plan, and accumulated observation of agent interactions with the environment, please generate all possible step-by-step global plans, which serve as high-level, natural guidance to assist in planning. Maintain the plan for all steps preceding the execution step index, while selectively modifying the plan for steps following the execution step index.

The action list you can take:

1. turn right
2. turn left
3. move forward
4. go to <obj> <id>
5. pick up <obj> <id>
6. go through <door> <id>: <door> must be an open door.
7. toggle and go through <door> <id>: <door> can be a closed door or a locked door. If you want to open a locked door, you need to carry a key that is of the same color as the locked door.
8. toggle: there is a closed or locked door right in front of you and you can toggle it.

where <obj> and <id> correspond to objects and index number.

#### # Task

{task}

#### # Previous Global Plan

{global\_plan} [optional]

#### # Execution Step Index

{execution\_step\_index}

#### # Accumulated Observation

{observation} [optional]

#### Output Format:

```
```json
["
Step 1: ...
```

Step 2: ...

```
...
"/, ...]
,,,
```

Prompt: Global Plan Generation - MAZE

You are an expert maze solver. Your objective is to reach the goal in as few steps as possible. Based on the task description, the previous global plan, and accumulated observation of agent interactions with the environment, please generate all possible step-by-step global plans, which serve as high-level, natural guidance to assist in planning. Maintain the plan for all steps preceding the execution step index, while selectively modifying the plan for steps following the execution step index. Your objective is to reach the goal in as few steps as possible. When you move right, you increase your y position by 1. When you move down, you increase your x position by 1.

The action list you can take:

1. move up
2. move down
3. move left
4. move right

For instance, given the current environment state: The goal is at position 8, 6. Your current position is at position 1, 1. There are walls to your left, above you, below you. The index of already executed steps is 0. The possible global plans could be:

[“

Step 1: move right (from 1, 1 to 1, 2)
Step 2: move right (from 1, 2 to 1, 3)
Step 3: move right (from 1, 3 to 1, 4)
Step 4: move down (from 1, 4 to 2, 4)
Step 5: move down (from 2, 4 to 3, 4)
Step 6: move down (from 3, 4 to 4, 4)
Step 7: move down (from 4, 4 to 5, 4)
Step 8: move down (from 5, 4 to 6, 4)
Step 9: move down (from 6, 4 to 7, 4)
Step 10: move down (from 7, 4 to 8, 4)
Step 11: move right (from 8, 4 to 8, 5)
Step 12: move right (from 8, 5 to 8, 6)
”, ...]

Task

{task}

Previous Global Plan
{global_plan} [optional]

Execution Step Index
{execution_step_index}

Accumulated Observation
{observation} [optional]

Output Format:

```
```json
[
 "
 Step 1: ...
 Step 2: ...
 ...
 ", ...]
,,,
```

**Prompt: Global Plan Selection (for the *generate-then-select* strategy)**

You are given several global plans serving as high-level, natural guidance to assist in planning. Based on the task description, accumulated observation of agent interactions with the environment, and the current index of execution step, please select the most suitable global plan from all available global plans for task completion.

When you select the global plan, consider evaluating the following aspects to identify the optimal choice based on comprehensive criteria:

1. **Correctness**: Does the global plan correctly and accurately address the task requirements?
2. **Executability**: Is the global plan clearly structured, easy to interpret, and are the individual steps logically sound and actionable?
3. **Standardization**: Does the global plan adhere to a consistent and standardized format?

**# Task**  
{task}

**# Available Global Plans**  
{global\_plans}

**# Execution Step Index**  
{execution\_step\_index}

**# Accumulated Observation**  
{observation} [optional]

**Prompt: Global Plan Quality Evaluation (for Equation (5))**

Please act as a professional guidance evaluator and judge the given global plan across the following three dimensions:

1. **Correctness**: Based on the environment's feedback on the agent's actions in response to the current global plan guidance, does the global plan accurately fulfill the task requirements?
2. **Executability**: Based on the agent's adherence to the global plan, is the global plan clear, easy to understand, and are the steps reasonable?
3. **Standardization**: Does the global plan adhere to a consistent and standardized format?

For each dimension, please score the global plan on a scale of 1 to 5, where 1 indicates poor performance and 5 indicates excellent performance, and explain the reason.

**# Task**  
{task}

**# Global Plan**  
{global\_plan}

**# Execution Step Index**  
{execution\_step\_index}

**# Accumulated Observation**  
{observation} [optional]

-----  
**Output Format:**

```
```json
{
  "correctness_score": xxx,
  "correctness_reason": "...",
  "executability_score": xxx,
  "executability_reason": "...",
  "standardization_score": xxx,
  "standardization_reason": "...
}
,,,
```

Prompt: Environmental Feedback

Based on the task description and the reference agent-environment interaction in which the agent has finally accomplished the task, please generate the environmental feedback for the agent's action and determine whether the current action has reached the final goal. If the agent's action has reached the fi-

nal goal, please output “Task Completed!”; else, the feedback should be in the following format: “Observation: ...”

Task
{task}

Reference Interaction
{ref_interaction}

Previous Observation
{observation} [optional]

Agent Action
{agent_action}

2. Think when necessary, but prioritize direct action wherever possible throughout the process.

Example
{example}

Task
{task}

Global Plan
{global_plan}

Previous Observation
{observation} [optional]

Prompt: Execution Generation - ALFWorld

Interact with a household to solve a task. Imagine you are an intelligent agent in a household environment and your target is to perform actions to complete the task goal. At the beginning of your interactions, you will be given the detailed description of the current environment and your goal to accomplish. For each of your turn, you will be given the observation of the last turn. You should choose from two actions: “Thought” or “Action”. If you choose “Thought”, you should first think about the current condition and plan for your future actions, and then output your action in this turn. Your output must strictly follow this format: “Thought: your thoughts. Action: your next action”; If you choose “Action”, you should directly output the action in this turn. Your output must strictly follow this format: “Action: your next action”.

For house holding task, the action list you can take:

1. go to recep
2. task obj from recep
3. put obj in/on recep
4. open recep
5. close recep
6. toggle obj recep
7. clean obj with recep
8. heat obj with recep
9. cool obj with rece

where obj and recep correspond to objects and receptacles.

Reminder:

1. The action is restricted to those listed as available. Actions not included in the provided list are considered invalid.

Prompt: Execution Generation - IQA

Imagine you are an intelligent agent in a dynamic visual environment and your target is to perform actions to complete the task goal. At the beginning of your interactions, you will be given the detailed description of the current environment and your goal to accomplish. For each of your turn, you will be given the observation of the last turn. You should choose from two actions: “Thought” or “Action”. If you choose “Thought”, you should first think about the current condition and plan for your future actions, and then output your action in this turn. Your output must strictly follow this format: “Thought: your thoughts. Action: your next action”; If you choose “Action”, you should directly output the action in this turn. Your output must strictly follow this format: “Action: your next action”.

The action list you can take:

1. move ahead
2. turn left
3. turn right
4. open obj
5. answer [True]/[False]

where obj correspond to objects.

Reminder:

1. The action is restricted to those listed as available. Actions not included in the provided list are considered invalid.
2. Think when necessary, but prioritize direct action wherever possible throughout the process.

Example
{example}

Task
{task}

Global Plan
{global_plan}

Previous Observation
{observation} [optional]

Prompt: Execution Generation - TextCraft

You are given a few useful crafting recipes to craft items in Minecraft. Crafting commands are of the format “craft [target object] using [input ingredients]”. Every round I will give you an observation, you have to respond to an action based on the state and instruction. You should choose from two actions: “Thought” or “Action”. If you choose “Thought”, you should first think about the current condition and plan for your future actions, and then output your action in this turn. Your output must strictly follow this format: “Thought: your thoughts. Action: your next action”; If you choose “Action”, you should directly output the action in this turn. Your output must strictly follow this format: “Action: your next action”. For your “Action”, you can “get” an object (ingredients) from the inventory or the environment, look up the game “inventory” by inventory, or “craft” (target) using any of the crafting commands. You can use ONLY these crafting commands provided, do not use your own crafting commands. However, if the crafting command uses a generic ingredient like “planks”, you can use special types of the same ingredient e.g. dark oak “planks” in the command instead. For any other natural language or thoughts, use prefix “Thought:”.

Reminder:

1. The action is restricted to those listed as available. Actions not included in the provided list are considered invalid.
2. Think when necessary, but prioritize direct action wherever possible throughout the process.

Example
{example}

Crafting Commands and Goal
{task}

Global Plan
{global_plan}

Previous Observation

{observation} [optional]

Prompt: Execution Generation - Wordle

You are an expert wordle player. Welcome to the game of Wordle. Your objective is to guess a hidden 5 letter word. You have 6 attempts to guess it correctly and you should try to guess it in as few attempts as possible. When guessing the word, you should format your word as a space separated sequence of letters, like “s h i r e” for example. After guessing the word, you will receive feedback from the game environment in the form of a sequence of 5 space separated letters like “b y g g b”, where each letter indicates some information about the hidden word. The environment will return one of three letters - “b”, “g”, or “y” – for each letter in the word you guessed. Here is the meaning of each letter:

- “b”: If the environment returns a “b”, it means that the letter at that position in your guessed word is not in the hidden word.
- “y”: If the environment returns a “y”, it means that the letter at that position in your guessed word is in the hidden word but is not in the correct position.
- “g”: If the environment returns a “g”, it means that the letter at that position in your guessed word is in the hidden word and is in the correct position.

For each of your turn, you will be given the observation of the last turn. You should choose from two actions: “Thought” or “Action”. If you choose “Thought”, you should first think about the current condition and plan for your future actions, and then output your action in this turn. Your output must strictly follow this format: “Thought: your thoughts. Action: your next action”; If you choose “Action”, you should directly output the action in this turn. Your output must strictly follow this format: “Action: your next action”.

Reminder:

1. The output format of the action should be a sequence of 5 individual letters, each separated by a space, such as “s h i r e”. Any other formats are considered invalid.
2. Think when necessary, but prioritize direct action wherever possible throughout the process.

Example
{example}

Task
{task}

Global Plan

{global_plan}

Previous Observation

{observation} [optional]

Prompt: Execution Generation - BabyAI

You are an exploration master that wants to finish every goal you are given. You are placed in a room and you need to accomplish the given goal with actions. For each of your turn, you will be given the observation of the last turn. You should choose from two actions: "Thought" or "Action". If you choose "Thought", you should first think about the current condition and plan for your future actions, and then output your action in this turn. Your output must strictly follow this format: "Thought: your thoughts. Action: your next action"; If you choose "Action", you should directly output the action in this turn. Your output must strictly follow this format: "Action: your next action".

The action list you can take:

1. turn right
2. turn left
3. move forward
4. go to <obj> <id>
5. pick up <obj> <id>
6. go through <door> <id>: <door> must be an open door.
7. toggle and go through <door> <id>: <door> can be a closed door or a locked door. If you want to open a locked door, you need to carry a key that is of the same color as the locked door.
8. toggle: there is a closed or locked door right in front of you and you can toggle it.

where <obj> and <id> correspond to objects and index number.

Reminder:

1. The action is restricted to those listed as available. Actions not included in the provided list are considered invalid.
2. Think when necessary, but prioritize direct action wherever possible throughout the process.

Example

{example}

Task

{task}

Global Plan

{global_plan}

Previous Observation

{observation} [optional]

Prompt: Execution Generation - MAZE

You are an expert maze solver. Your objective is to reach the goal in as few steps as possible. At each step you will be given information about where the goal is, your current position, and the walls that surround you. You should choose from two actions: "Thought" or "Action". If you choose "Thought", you should first think about the current condition and plan for your future actions, and then output your action in this turn. Your output must strictly follow this format: "Thought: your thoughts. Action: your next action"; If you choose "Action", you should directly output the action in this turn. Your output must strictly follow this format: "Action: your next action". Specifically, when you move right, you increase your y position by 1. When you move down, you increase your x position by 1.

The action list you can take:

1. move up
2. move down
3. move left
4. move right

Reminder:

1. The action is restricted to those listed as available. Actions not included in the provided list are considered invalid.
2. Think when necessary, but prioritize direct action wherever possible throughout the process.

Example

{example}

Task

{task}

Global Plan

{global_plan}

Previous Observation

{observation} [optional]

Prompt: Adherence Degree Judgment (for Equation (3))

You are an expert in agent tasks. You are tasked with evaluating the agent's execution of a given global plan. Specifically, you are to assess the degree of compliance between the agent's actions and the strategic guidance outlined in the global plan. Rate it from 0 to 2 points, and explain the reason.

2 Point Answer Criteria:

The agent's execution strictly adheres to the guidance provided in the global plan. All actions are logically aligned with the plan's objectives and are carried out as instructed.

1 Point Answer Criteria:

The agent's execution demonstrates a partial alignment with the global plan, allowing for minor deviations. For example, in cases where the plan suggests the use of multiple tools, the agent may use at least one relevant tool to support the execution, as long as it does not contradict the overall guidance.

0 Point Answer Criteria:

The agent's execution departs or contradicts the global plan, or contains garbled characters, format errors, disorder, and irrelevant information.

Task

{task}

Global Plan

{global_plan}

Execution Step Index

{execution_step_index}

Agent Action

{agent_action}

Output Format:

```
```json
{
 "score": xxx,
 "reason": "... "
}
```
```

Prompt: E2E Performance Judgment (for Equation (4))

You are an expert in agent tasks. Please evaluate the end-to-end (E2E) performance of the agent during its interaction with a given environment. The goal is

to assess whether the agent accomplishes the target task efficiently and directly, without unnecessary detours or redundancies. Rate it from 0 to 2 points, and explain the reason.

2 Point Answer Criteria:

1. The agent successfully completes the task in a direct and efficient manner.
2. There are no unnecessary steps or redundant actions in the interaction trajectory.

1 Point Answer Criteria:

1. The task is ultimately completed, but the process includes some level of redundancy or unintended topic drift.
2. While the final objective is met, there may be deviations from the optimal path.

0 Point Answer Criteria:

1. The agent fails to achieve the final task objective.
2. Contains significant deviations, errors, or inability to progress towards the goal.

Task

{task}

Agent-Environment Interaction

{accumulated_context}

Reference Interaction

{ref_interaction}

Output Format:

```
```json
{
 "score": xxx,
 "reason": "... "
}
```
```

Case Studies

For agent tasks involving multi-step decision-making, generating a global plan to guide the execution of each step is crucial. This is because models may forget the previous context after executing multiple steps, leading to redundant actions or failure to accomplish the task. As shown in Figure 5, the red annotations indicate redundant interaction trajectories during the problem-solving process. When the agent has already moved three steps to the right, it forgets that the red ball should be directly on its left and continues to move forward, resulting in a large amount of redundant executions. In contrast, as depicted in Figure 6, with the guidance of a global plan, the agent can clearly recognize its relative position of the target, thereby efficiently completing the task.

Task Instruction

You are placed in a room and you need to accomplish the given goal with actions. Your goal is to: go to the red ball. In front of you in this room, you can see several objects: There is a red ball 1 4 steps in front of you and 3 steps to your right. The room has walls around you. You are facing a wall 6 steps away. You are not carrying anything.

ReAct

Thought: The red ball is 4 steps in front of me and 3 steps to my right, which means I need to turn right to reach it. **Action:** turn right

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 5 steps away. You are not carrying anything.

Thought: I need to keep moving forward to find the red ball. **Action:** move forward

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 4 steps away. You are not carrying anything.

Thought: I still need to move forward to get closer to the red ball. **Action:** move forward

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 3 steps away. You are not carrying anything.

Thought: I need to continue moving forward to get closer to the red ball. **Action:** move forward

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 2 steps away. You are not carrying anything.

Thought: The red ball should be nearby, I should be getting closer. **Action:** move forward

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 1 steps away. You are not carrying anything.

Thought: I can't move forward anymore. I should turn right to find the red ball. **Action:** turn right

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 1 steps away. You are not carrying anything.

Thought: I need to turn right again to find the red ball. **Action:** turn right

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 6 steps away. You are not carrying anything.

Thought: I need to turn left to move towards the red ball. **Action:** turn left

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 1 steps away. You are not carrying anything.

Thought: The red ball should be close by, I should turn left to find it. **Action:** turn left

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 1 steps away. You are not carrying anything.

Thought: I need to turn left again to keep searching for the red ball. **Action:** turn left

Observation: In front of you in this room, you can see several objects: There is a red ball 1 4 steps in front of you and 1 steps to your left. The room has walls around you. You are facing a wall 6 steps away. You are not carrying anything.

Thought: The red ball is 4 steps in front of me and 1 steps to my left, which means I need to turn left to reach it. **Action:** turn left

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 6 steps away. You are not carrying anything.

Thought: I need to keep moving forward to find the red ball. **Action:** move forward

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 5 steps away. You are not carrying anything.

Thought: I need to turn right to move towards the red ball. **Action:** turn right

Observation: In front of you in this room, you can see several objects: There is a red ball 1 4 steps in front of you. The room has walls around you. You are facing a wall 6 steps away. You are not carrying anything.

Thought: I finally found the red ball! I need to go directly to the red ball. **Action:** go to red ball 1

Figure 5: Case study of ReAct (Yao et al. 2023) on BabyAI (Chevalier-Boisvert et al. 2019).

Task Instruction

You are placed in a room and you need to accomplish the given goal with actions. Your goal is to: go to the red ball. In front of you in this room, you can see several objects: There is a red ball 14 steps in front of you and 3 steps to your right. The room has walls around you. You are facing a wall 6 steps away. You are not carrying anything.

PilotRL (AdaPlan)

Global Plan:

Step 1: Turn right to face the direction where the red ball is offset.

Step 2 : Move forward as the 1st step toward the red ball's lateral position.

Step 3 : Move forward as the 2nd step toward the red ball's lateral position.

Step 4 : Move forward as the 3rd step toward the red ball's lateral position.

Step 5: Turn left to face the original front direction.

Step 6: Go to the red ball 1.

Thought: The global plan starts with turning right to face the red ball's lateral direction. Since the red ball is 3 steps to my right, turning right will align me with that direction. **Action:** turn right

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 5 steps away. You are not carrying anything.

Thought: I need to keep moving forward to find the red ball. According to Step 2 of the global plan, I need to take the first step toward the red ball's lateral position. Moving forward now will reduce the distance to the red ball. **Action:** move forward

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 4 steps away. You are not carrying anything.

Thought: Step 3 requires the second step toward the red ball's lateral position. Continuing to move forward will bring me closer. **Action:** move forward

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 3 steps away. You are not carrying anything.

Thought: Following Step 4, this is the third step toward the red ball's lateral position. Moving forward should align me with the red ball's lateral position. **Action:** move forward

Observation: In front of you in this room, you can see several objects: The room has walls around you. You are facing a wall 2 steps away. You are not carrying anything.

Thought: Step 5 requires turning left to face the original front direction, which is where the red ball is now located ahead. Turning left will let me face the red ball directly. **Action:** turn left

Observation: In front of you in this room, you can see several objects: There is a red ball 14 steps in front of you. The room has walls around you. You are facing a wall 6 steps away. You are not carrying anything.

Thought: I finally found the red ball! I need to go directly to the red ball. **Action:** go to red ball 1

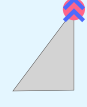
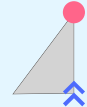
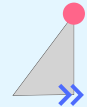
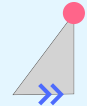
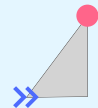


Figure 6: Case study of PilotRL (AdaPlan) on BabyAI (Chevalier-Boisvert et al. 2019).