

000 PIOTRL: TRAINING LANGUAGE MODEL AGENTS 001 VIA GLOBAL PLANNING-GUIDED PROGRESSIVE RE- 002 INFORCEMENT LEARNING 003 004

006 **Anonymous authors**

007 Paper under double-blind review

011 ABSTRACT

013 Large Language Models (LLMs) have shown remarkable advances in the ability to
014 tackle agent-oriented tasks. Despite their potential, existing work faces challenges
015 when deploying LLMs in agent-based environments. The widely adopted agent
016 paradigm ReAct centers on the integration of single-step reasoning with immedi-
017 ate action execution, which limits its effectiveness in complex tasks that require
018 long-term strategic planning. Furthermore, the coordination between the planner
019 and executor during problem-solving is also a critical factor to consider in agent
020 design. Additionally, current approaches predominantly rely on supervised fine-
021 tuning, which often leads models to memorize established task completion trajec-
022 tories, thereby restricting their generalization ability when confronted with novel
023 problem contexts. To address these challenges, we introduce an adaptive global
024 plan-based agent paradigm **AdaPlan**, aiming to synergize high-level explicit guid-
025 ance with execution to support effective long-horizon decision-making. Based on
026 the proposed paradigm, we further put forward **PilotRL**, a global planning-guided
027 training framework for LLM agents driven by progressive reinforcement learning.
028 We first develop the model’s ability to follow explicit guidance from global plans
029 when addressing agent tasks. Subsequently, on the basis of this foundation, we
030 focus on optimizing the quality of generated plans. Finally, we conduct joint
031 optimization of the model’s planning and execution coordination. Extensive ex-
032 periments indicate that **PilotRL** could achieve state-of-the-art performances, with
033 LLaMA3.1-8B-Instruct + PilotRL surpassing closed-sourced GPT-4o by 3.60%,
034 while showing a more substantial gain of 55.78% comparing to GPT-4o-mini at a
035 comparable parameter scale.

036 1 INTRODUCTION

038 An *agent* can be defined as an entity capable of perceiving its environment, making decisions, and
039 executing actions in pursuit of predefined or adaptive goals (Wooldridge & Jennings, 1995; Maes,
040 1995; Jennings et al., 1998). The state-of-the-art Large Language Models (LLMs), such as GPT-
041 4 (Achiam et al., 2023) and Gemini (Team et al., 2023), have exhibited strong agent capabilities,
042 including instruction following, reasoning, and programming, which inspires widespread efforts to
043 develop autonomous agent systems with LLMs serving as central cognitive controllers (Song et al.,
044 2023; Sumers et al., 2023). Nevertheless, considering the high financial costs and safety risks of
045 close-sourced proprietary models (Li et al., 2023; Yuan et al., 2023), recent efforts have been shifted
046 to improve such agent capabilities in open-sourced models as effective alternatives (Chen et al.,
047 2024; Song et al., 2024; Fu et al., 2025).

048 Despite their potential, existing works face some limitations: **(C1) Limited Contextual Awareness**
049 **of ReAct**: While the ReAct paradigm (Yao et al., 2023) is a general foundation of modern agen-
050 tic systems, it lacks insight into the overarching context. The reasoning component (generated as
051 “thought”) focuses purely on immediate action, which limits its effectiveness in complex tasks re-
052 quiring sequential execution. **(C2) Insufficient Coordination between Planning and Executing**:
053 Although recent studies have incorporated planning into agent-based problem-solving process (Er-
dogan et al., 2025; Xiong et al., 2025), they design the planner and executor in isolation, leading

to potential mismatches between the two components. As a result, the plans generated by the planner may not be effectively followed by the executor, undermining overall task performance. **(C3)** **Deficient Generalization of SFT:** Extensive research has been devoted to enhancing the agent capabilities of models through supervised fine-tuning (SFT) (Deng et al., 2023; Zeng et al., 2024). However, recent studies indicate that SFT tends to lead models to memorize task-specific heuristics rather than acquiring generalizable capabilities applicable to new scenarios (Chu et al., 2025).

To address these challenges, we introduce **PilotRL**, which is a global plan-driven reinforcement learning framework for the training of LLM agents. For **C1** and **C2**, we propose the adaptive global plan-based paradigm *AdaPlan* to guide the agent through complex tasks as a pilot, where global plans are dynamically generated and continuously updated throughout the execution process. The global planner and executor are implemented within a unified model to enhance their coordination and mutual adaptability. For **C3**, we employ reinforcement learning (RL) for its high effectiveness at enhancing generalizable knowledge in LLMs (Jaech et al., 2024; Guo et al., 2025; Team et al., 2025), the training process of which can be divided into three stages: **(1) Stage 1: Executor Enhancement.** We begin by developing the executor’s instruction adhesion to the global plan when addressing agent tasks. **(2) Stage 2: Global Planner Cultivation.** Building upon the global plan following, we optimize the global planner to improve the coordination. Finally, we refine the coordination by enhancing their collaborative performance in

Contributions. The main contributions can be summarized as follows:

- *Paradigm Innovation.* We introduce an adaptive global plan-based agent paradigm, AdaPlan, to synergize high-level reasoning with executing for long-horizon decision-making. By integrating both the global planner and executor in a unified model, our approach enables more effective coordination and improved end-to-end performance.
- *Training Framework Advancement.* Based on AdaPlan, we propose PilotRL, a global planning-guided progressive reinforcement learning framework designed for enhancing the agent capabilities of models via a three-stage process.
- *Performance and Effectiveness.* Extensive experiments indicate the superiority of PilotRL. Notably, models trained with PilotRL even surpasses closed-sourced proprietary models for agent tasks, achieving average improvements over GPT-4o and GPT-4o-mini by 2.35% and 53.90%.

2 PILOTRL

Assuming the scenario where an agent interacts with an environment for task solving, we present a detailed overview of our proposed PilotRL framework in this section.

2.1 ADAPLAN: ADAPTIVE GLOBAL PLANNING

While ReAct (Yao et al., 2023) is effective in many interactive agent tasks, its reliance on single-step reasoning and immediate action generation limits its capability in scenarios that require extended planning and coherent decision-making. To address this, we introduce the *AdaPlan* paradigm, which focuses on the adaptively generated and refined global plan throughout the task-solving process.

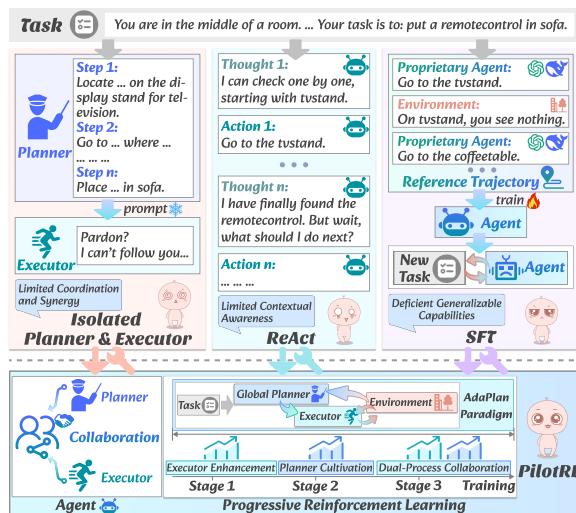


Figure 1: Comparison of PilotRL (*bottom*) with existing methods (*top*) for agent task completion.

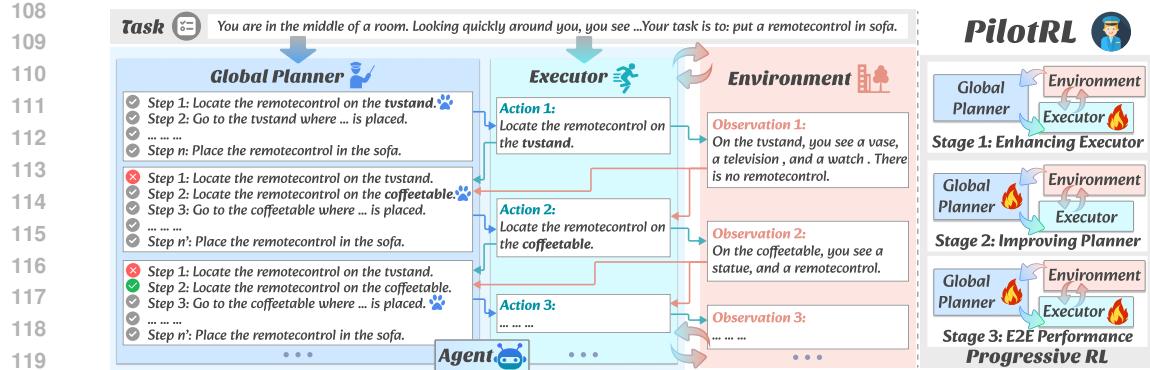


Figure 2: Overview of **PilotRL**. (Left) In **AdaPlan** paradigm, the *global planner* begins by processing the task instruction and generates an initial high-level plan for guidance, which is then passed to the *executor* for action generation. The observation from the *environment* is then fed back to both the *executor* for subsequent action generation and to the *global planner* for plan adaptation in response to changes or unexpected outcomes. (Right) The three-stage training process of **progressive RL**.

As shown in the left part of Figure 2, the agent architecture consists of two key components: the *global planner* and the *executor*. For a given task instruction G and the initial context $\mathcal{C}^{(0)}$, the *global planner* first generates the global plan $\mathcal{P}^{(0)} = [p_1^{(0)}, p_2^{(0)}, \dots, p_{N_0}^{(0)}]$ consisting of N_0 steps, where $p_i^{(0)}$ represents the recommended action for the *executor* at step i under the current planning strategy. At each time step t ($t \geq 1$), the *executor* takes an action $a^{(t)} \in \mathcal{A}$ based on: (1) the previous accumulated context $\mathcal{C}^{(t-1)} = \{(a^{(j)}, o^{(j)})\}_{j=0}^{t-1}$, where $o \in \mathcal{O}$ refers to observation from the environment, and (2) the guidance from the current global plan $\mathcal{P}^{(t-1)}$. Subsequently, it receives the resulting observation $o^{(t)} \in \mathcal{O}$, and the current turn of agent-environment interaction $(a^{(t)}, o^{(t)})$ is incorporated into the accumulated context $\mathcal{C}^{(t)}$ of the execution step t . The global plan $\mathcal{P}^{(t-1)}$ is then iteratively refined according to the task goal G and the accumulated context $\mathcal{C}^{(t)}$ to facilitate the next execution, resulting in $\mathcal{P}^{(t)}$. Each $p_i^{(t-1)}$ in the original $\mathcal{P}^{(t-1)}$ is updated as follows:

$$p_i^{(t)} = \begin{cases} p_i^{(t-1)}, & \text{if } i \leq t \\ \pi(p_i^{(t)} | G, \mathcal{C}^{(t)}, \mathcal{P}^{(t-1)}, i), & \text{if } i > t \end{cases} \quad (1)$$

where π is the adaptation policy of the global plan generator.

By dynamically updating the global plan based on real-time feedback derived from the executor-environment interactions, the agent can promptly assess the validity and efficiency of the current planning strategy and make necessary adjustments accordingly. Furthermore, in cases where the executor deviates from the prescribed plan, the global planner can adaptively revise the course of action to guide the executor toward more effective task execution.

2.2 PROGRESSIVE REINFORCEMENT LEARNING

Within the global planning-driven agent paradigms, two key factors influence the overall performance: the quality of the generated global plans, and the degree to which the executor adheres to the plan’s directives when interacting with the environment. Accordingly, we employ a three-stage pipeline for training, as shown in the right part of Figure 2.

2.2.1 STAGE 1: ENHANCING THE INSTRUCTION ADHERENCE IN EXECUTOR

The ability to comply with the guidance of the global planner is foundational to the entire global planning-guided agent paradigm. Therefore, our focus lies on improving the executor’s capacity to follow existing global plans as well as acquire a thorough understanding of the action space \mathcal{A} in the initial training stage. Here we utilize the frontier model, e.g., DeepSeek-V3 (Liu et al., 2024), for the provision of each global plan. Specifically, for each time step t :

- **Plan Generation:** We first prompt the model to generate all possible global plans based on the specified task goal G and accumulated contextual information $\mathcal{C}^{(t)}$ to provide a comprehensive set of potential candidates.
- **Plan Selection:** Following this, the model evaluates each of these candidate plans across the dimensions of correctness, executability, format validity, etc., and selects the most suitable one to guide the executor’s actions.

In general, the design of the reward function in Stage 1 is *the sum of normalized components: format, adherence degree, and end-to-end (E2E) performance*.

Format. The model is required to produce its outputs according to a predefined output paradigm. Specifically: (1) All responses should be chosen from the two actions, “Thought” or “Action”, and must strictly align with the formats of “Thought: ... Action: ...” or “Action: ...”. (2) The output must be produced in a readable format, without distorted or illegible characters, and then the environmental feedbacks are encapsulated within `<observation>...</observation>` tags. Based on the above requirements, the format reward is defined as:

$$\mathcal{R}_{format} = \begin{cases} 1, & \text{if the format is correct} \\ 0, & \text{if the format is incorrect} \end{cases} \quad (2)$$

Adherence Degree. This aspect constitutes a core component in fostering the executor’s compliance with the global plan during Stage 1. Here we employ a frontier model (e.g., DeepSeek-V3) as the evaluator to score the generated actions. It assesses whether the model’s output semantically aligns with the current step of the global plan. Actions are assigned a score of 2 for fully compliant, 1 for partially compliant (e.g., for suggested actions that require the invocation of multiple tools, at least one tool is utilized to support the execution), and 0 for noncompliant actions:

$$\mathcal{R}_{adherence} = \begin{cases} 2, & \text{if completely compliant} \\ 1, & \text{if partially compliant} \\ 0, & \text{if noncompliant} \end{cases} \quad (3)$$

End-to-End (E2E) Performance. The measurement of the first two components concentrates solely on individual execution, rather than assessing the holistic interaction between the agent and the environment. However, in real-world interactions, the problem-solving process may exhibit trajectory redundancy or unintended topic drift, leading to unpredictable deviations from the intended workflow. Therefore, it is essential to obtain a comprehensive, end-to-end view of agent performance in order to assess whether the current interaction trajectory aligns with the expected behavior, and to ensure that the target task is accomplished efficiently and directly, without unnecessary detours.

$$\mathcal{R}_{E2E} = \begin{cases} 2, & \text{if accomplished efficiently} \\ 1, & \text{if accomplished with redundancy} \\ 0, & \text{if unaccomplished} \end{cases} \quad (4)$$

We utilize DeepSeek-V3 to evaluate the end-to-end (E2E) performance \mathcal{R}_{E2E} . The agent-environment interactions receive a score of 2 if the task is accomplished in a direct and efficient manner without process redundancy. A score of 1 is assigned if the final task is completed but the interaction involves trajectory redundancy or topic drift. If the agent fails to achieve the final task objective, it is given a score of 0.

2.2.2 STAGE 2: CULTIVATING THE CAPACITY OF GLOBAL PLANNER

Following the initial training stage, the agent has acquired a foundational paradigm for global plan following and action execution. In this stage, we shift our focus to enhancing the agent’s ability to generate global plans. In generating the global plan, we adopt a *generate-then-select* strategy similar to that used in Stage 1 with the frontier model, which enhances the quality of the global plan ultimately used for explicit guidance, leading to more effective and coherent decision-making. Specifically, all feasible global plans that could potentially solve the given task are first generated, and then the most appropriate one is selected from this pool of candidates. The reward function design in Stage 2 is *the sum of normalized components: format, end-to-end (E2E) performance, and global plan quality*, with the first two already formally defined in Equation (2) and Equation (4).

216 **Global Plan Quality** When evaluating the quality of the generated global plan, we consider
 217 three primary dimensions: *correctness*, *executability*, and *standardization*. (1) *Correctness* assesses
 218 whether the plan effectively leads to the fulfillment of the task objectives. (2) *Executability* eval-
 219 uates the clarity and ease with which the agent can adhere to the instructions, as indicated by the
 220 alignment of the executor’s action with the global planner’s directives. (3) *Standardization* checks
 221 whether the generated instructions conform to a consistent and well-defined format. The quality
 222 score of the global plan is calculated as follows:

$$R_{\text{planning}} = R_{\text{correct}} + R_{\text{execute}} + R_{\text{standard}} \quad (5)$$

223 where $R_{\text{correct}}, R_{\text{execute}}, R_{\text{standard}} \in \{x \in \mathbb{Z} \mid 1 \leq x \leq 5\}$, with 5 indicating the best perfor-
 224 mance. We use DeepSeek-V3 as the evaluator to score each dimension.
 225

227 2.2.3 STAGE 3: ORCHESTRATING THE END-TO-END (E2E) PERFORMANCE

228 Having separately enhanced the model’s capabilities in both generating and complying with global
 229 plans in earlier stages, Stage 3 focuses on strengthening the coordination between the global planner
 230 and the executor, i.e., the joint optimization of our global planning-driven agent paradigm *AdaPlan*.
 231 The reward function at this stage is the *sum of normalized format* and *end-to-end (E2E) perfor-*
 232 *mance*, which directly prioritizes comprehensive performance of the ultimate task objective.
 233

234 3 EXPERIMENTS

235 3.1 EXPERIMENTAL SETUP

236 **Datasets.** During *training*, we collect data from the training splits of four datasets: ALF-
 237 World (Shridhar et al., 2021), IQA (Gordon et al., 2018), TextCraft (Prasad et al., 2024), and Wor-
 238 dle (Abdulhai et al., 2023). Our *evaluation* is conducted on six benchmarks. We employ the test
 239 splits of ALFWORLD, IQA, TextCraft, and Wordle for in-domain (ID) assessment, and the full dataset
 240 samples of MAZE (Abdulhai et al., 2023) and BabyAI (Chevalier-Boisvert et al., 2019) for out-of-
 241 domain (OOD) scenarios. We collected data from prior work (Song et al., 2024; Xi et al., 2024), and
 242 use only the task instructions and their corresponding final answers for RL-related training and eval-
 243 uation, with the overall statistics and details of the datasets described in Table 5 and Section B.1.
 244 In this work, we adopt the *LLM-as-Judge* (Zheng et al., 2023; Gu et al., 2024) paradigm to verify
 245 the model’s end-to-end (E2E) performance, including (1) the task completion rates, and (2) the
 246 efficiency of interaction trajectories, and then calculate the average scores as the evaluation metric.
 247

248 **Models and Implementation.** We validate the effectiveness of PilotRL across different base
 249 and instruction-tuned models, including Qwen2.5-7B-Instruct (Yang et al., 2024), LLaMA3.1-8B-
 250 Instruct (Dubey et al., 2024), and Qwen3-8B (Yang et al., 2025). The *reinforcement learning*
 251 (*RL*) framework is built on verl (Sheng et al., 2025) with Group Relative Policy Optimization
 252 (GRPO) (Shao et al., 2024) as the learning algorithm. The total training dataset contains 5725
 253 samples. Each sample undergoes 16 rollouts, with a training batch size of 256 and a rollout batch
 254 size of 64. The total number of training epochs is set to 4, with 1 epoch allocated to Stage 1, 2 epochs
 255 to Stage 2, and an additional 1 epoch dedicated to Stage 3. The learning rate is set at 1e-6. Following
 256 the approach proposed by Sun et al. (2025), we employ the frontier model DeepSeek-V3 to simulate
 257 real-world environmental behaviors. Notably, in our training setup, the environmental observation
 258 \mathcal{O} is concatenated into the interaction process, which are not generated by the training policy. To
 259 prevent these segments from influencing gradient updates, we apply masking during loss calculation,
 260 where we mask out all content enclosed within `<observation>...</observation>` tags.
 261 When conducting *supervised fine-tuning (SFT)* as baseline competitors, we utilized a learning rate
 262 scheduler featuring linear warm-up and cosine decay, peaking at a learning rate of 2e-5, alongside a
 263 warmup ratio of 0.03 and a weight decay of 0.0 and a batch size of 256 for 4 epochs.
 264

265 **Baselines.** We compare PilotRL with the following baselines: (1) We employ GPT-4o and GPT-
 266 4o-mini (Hurst et al., 2024) as the *Close-Sourced Models* competitors. (2) *Open-Sourced Agent-
 267 Specific Models* include Agent-FLAN-7B (Chen et al., 2024), LLaMA-xLAM-2-8B-fc-r (Zhang
 268 et al., 2024a) and DeepResearcher-7B (Zheng et al., 2025). (3) The simplest baseline is *Naive Re-
 269 sponse*, where the model generates responses directly without any training or prompting strategies.
 (4) *ReAct* (Yao et al., 2023) is the common agent paradigm that prompts agents to integrate single-
 step reasoning with immediate action execution. (5) *MPO* (Xiong et al., 2025) acts as an external

270
 271 Table 1: Comparison of PilotRL with baselines. “w/o Plan.” indicates whether the inference
 272 paradigm includes global planning as a mechanism for providing explicit guidance. The best and
 273 second best of each model are in **bold** and underlined.

Backbone Model	Method	w/o Plan.	ALFWorld	IQA	TextCraft	Wordle	BabyAI	MAZE	Avg.
<i>In-Domain (ID)</i>									
<i>Close-Sourced Models</i>									
GPT-4o GPT-4o-mini	– –	✗ ✗	75.83 52.35	66.59 40.32	68.50 46.74	78.65 42.51	57.87 43.96	60.42 34.36	67.98 45.21
<i>Open-Sourced Agent-Specific Models</i>									
Agent-FLAN-7B LLaMA-xLAM-2-8B-fc-r DeepResearcher-7B	– – –	✗ ✗ ✗	70.54 50.38 58.36	57.62 53.74 62.87	24.66 46.15 55.58	22.28 48.52 47.17	24.39 54.26 52.75	28.93 36.57 40.82	38.07 48.27 52.93
<i>Open-Sourced Base / Instruct Models</i>									
Qwen2.5-7B-Instruct	Naive Response	✗	48.78	35.40	30.35	34.72	40.39	33.80	37.24
	ReAct	✗	52.15	37.57	34.46	40.43	44.08	37.52	41.04
	+ MPO	✓	67.31	58.64	52.28	56.76	53.85	49.67	56.42
	SFT	✓	67.53	63.35	73.10	74.64	55.68	46.92	63.54
	Vanilla RL	✗	65.49	64.78	70.76	71.28	58.62	50.59	63.59
	PilotRL (ours)	✓	70.80	67.84	<u>75.37</u>	<u>77.69</u>	61.56	57.93	68.53
LLaMA3.1-8B-Instruct	Naive Response	✗	35.63	38.56	38.22	36.40	46.17	30.64	37.60
	ReAct	✗	38.48	42.94	45.83	38.56	47.36	36.92	41.68
	+ MPO	✓	54.25	50.31	43.86	52.60	58.92	45.33	50.88
	SFT	✓	74.92	69.84	58.42	73.55	55.52	50.76	63.84
	Vanilla RL	✗	70.68	68.13	60.57	68.80	59.74	52.05	63.33
	PilotRL (ours)	✓	78.53	72.78	64.76	79.61	68.24	58.68	70.43
Qwen3-8B	Naive Response	✗	54.08	42.14	36.37	34.95	48.46	36.53	42.09
	ReAct	✗	62.56	50.58	44.62	41.60	54.35	42.68	49.40
	+ MPO	✓	65.42	54.67	46.25	48.79	56.81	39.50	51.91
	SFT	✓	64.73	62.75	63.16	75.83	59.67	49.25	62.57
	Vanilla RL	✗	68.47	70.29	<u>67.35</u>	<u>80.42</u>	<u>63.44</u>	<u>52.04</u>	<u>67.00</u>
	PilotRL (ours)	✓	72.51	<u>69.06</u>	71.48	83.65	65.28	56.62	69.77

297 plug-and-play planner that endows the model with meta-plans to provide explicit guidance during
 298 task execution. (6) We also perform *Supervised Fine-Tuning (SFT)* on models, a widely adopted
 299 training strategy in a series of previous works (Chen et al., 2024; Song et al., 2024; Xi et al., 2024;
 300 Zeng et al., 2024; Zhang et al., 2024b; Fu et al., 2025). Specifically, we utilize frontier models
 301 (e.g., DeepSeek-V3) to generate global plans that guide the execution of target tasks. (7) *Vanilla*
 302 *RL* is the naive reinforcement learning process that utilizes the Group Relative Policy Optimization
 303 (GRPO) (Shao et al., 2024) algorithm. In this setup, we utilize only the format and end-to-end (E2E)
 304 performance as the reward metrics. Details are discussed in Section B.2.

3.2 MAIN RESULTS

307 The main results of baselines and PilotRL are demonstrated in Table 1, and we show the methods
 308 based on models with base and instruction-tuned versions respectively. From the main results, we
 309 summarize the observations below.

310 **PilotRL is effective across different models.** Experimental results in Table 1 show that our PilotRL
 311 consistently outperforms other baseline approaches on both base and instruction-tuned models in
 312 terms of agent task completion. Compared to the *naive response*, PilotRL enhances the average
 313 downstream task performances by 78.51%. Remarkably, when compared to *open-sourced agent-*
 314 *specific models* such as DeepResearcher-7B, our approach achieves over 29.47% higher performance
 315 with the same backbone model of Qwen2.5-7B-Instruct. In comparison to the plug-and-play external
 316 planner *MPO*, our method achieves an average improvement of 31.10%, further highlighting the
 317 importance of tight coordination between the planner and executor in effectively solving agent-
 318 oriented tasks. Furthermore, open-sourced models enhanced with PilotRL demonstrate the potential
 319 to outperform *close-sourced proprietary models* in agent problem-solving. Specifically, models
 320 integrated with PilotRL achieve an average improvement of 2.35% over GPT-4o, while showing a
 321 more substantial gain of 53.90% over GPT-4o-mini at a comparable parameter scale.

322 **AdaPlan paradigm + RL boosts agent performances.** Here we focus on analyzing the performance
 323 of two baseline methods: *SFT* and *Vanilla RL*. The primary distinction between SFT and
 PilotRL lies in the training strategies, while the key difference between Vanilla RL and our method

324 is whether to incorporate the AdaPlan paradigm to provide global guidance for agent execution.
 325 As presented in Table 1, the average performance of SFT and Vanilla RL is quite similar on both
 326 Qwen2.5-7B-Instruct and LLaMA3.1-8B-Instruct. This suggests that the enhancement brought by
 327 global plan guidance in SFT is roughly on par with the incremental gain achieved through RL-based
 328 training. Specifically, for in-domain (ID) tasks, SFT outperforms Vanilla RL by a marginal average
 329 of 2.75%, whereas Vanilla RL achieves an average lead of 5.80% in out-of-domain (OOD) tasks. For
 330 reasoning-oriented models such as Qwen3-8B, which inherently possess a certain degree of multi-
 331 step reasoning and decision-making capabilities required for complex agent tasks, the performance
 332 gains from the AdaPlan paradigm are insufficient to offset the advantages of RL over SFT training.
 333 In contrast, PilotRL demonstrates robust performance gains across models with diverse character-
 334 istics, achieving consistent improvements over both SFT and Vanilla RL by 9.89% and 7.64%,
 335 respectively. These observations further highlight the importance of combining the global planning
 336 capabilities of the AdaPlan paradigm with RL training, as embodied in our PilotRL framework, for
 337 enhancing model performance in complex agent scenarios.
 338

4 ABLATIONS AND ANALYSIS

340 We conduct ablation studies on open-sourced models to highlight the contribution of each training
 341 stage and to assess the impact of their sequential order on PilotRL. Furthermore, we perform an in-
 342 depth analysis of PilotRL’s effectiveness, examining key aspects such as our AdaPlan paradigm for
 343 explicit guidance, the architecture of unified planner-executor, and the co-evolution of components.
 344

4.1 TRAINING STAGE ABLATION

Necessity of Progressive Training.

345 We aggregated the reward functions
 346 from all training stages to verify the im-
 347 portance of incrementally optimizing
 348 the planning and execution capabilities
 349 in a staged and progressive manner. Re-
 350 sults are presented in Table 2 (1 & 2
 351 & 3), where we observe a performance
 352 drop of 3.32% compared to our multi-
 353 stage training strategy (1 → 2 → 3).
 354 A primary cause of this performance
 355 drop lies in the intrinsic complexity
 356 and potential conflicts among heteroge-
 357 neous reward signals. Specifically, the
 358 planning-oriented and execution-driven
 359 components exert distinct behavioral
 360 demands on the model, which can lead
 361 to unstable policy updates during train-
 362 ing. For instance, in the early stages
 363 of training, the model may lack a suf-
 364 ficiently mature structure for guidance
 365 follow-up, making it difficult to accu-
 366 rately adhere to global plans. It results
 367 in conflicting gradient signals and ulti-
 368 mately reduces learning efficiency.
 369

370 **The Role of Each Stage.** To assess the contribution of each individual stage, we conduct three
 371 ablation studies by sequentially removing Stage 1, 2, and 3, respectively. The models are then
 372 evaluated on both in-domain (ID) and out-of-domain (OOD) benchmark tasks, with the results pre-
 373 sented in Table 2. To ensure a fair comparison and control for the impact of training data volume
 374 on performance, we fix the total number of training epochs at 4, which is consistent with the main
 375 experimental setup, and allocate 2 epochs to each of the remaining two stages for training.
 376

- 377 • Removing Stage 1. Stage 1 is designed to strengthen the models’ ability to follow instructions
 378 when performing agent tasks. As shown in Table 2 (2 → 3), the removal of Stage 1 results in

Table 2: Analysis of the training stages and sequential order. “Order” refers to the sequence of Stage 1, 2, and 3. “1 & 2 & 3” denotes a training setting in which the reward functions from all stages are applied simultaneously. We compute the *average* performance of the evaluated models across each benchmark. The best and second best scores are in **bold** and underlined.

Order	In-Domain	Out-of-Domain	Avg.
<i>Standard Pipeline</i>			
1 → 2 → 3	73.68	61.39	69.58
<i>Necessity of Progressive Training</i>			
1 & 2 & 3	71.64 (↓ 2.77%)	58.52 (↓ 4.68%)	67.27 (↓ 3.32%)
<i>The Role of Each Stage</i>			
2 → 3	70.82 (↓ 3.88%)	58.33 (↓ 4.98%)	66.66 (↓ 4.20%)
1 → 3	70.66 (↓ 4.10%)	58.39 (↓ 4.89%)	66.57 (↓ 4.33%)
1 → 2	72.21 (↓ 2.00%)	59.02 (↓ 3.86%)	67.81 (↓ 2.54%)
<i>Sequential Order of Stages</i>			
2 → 1 → 3	<u>72.79</u> (↓ 1.21%)	59.88 (↓ 2.46%)	68.48 (↓ 1.58%)

378 a performance drop of 4.20% in overall model performance. This decline occurs because Stage
 379 1 acts as the cornerstone for Stage 2. Without robust instruction-following behavior, the model
 380 struggles to adhere to the provided global plans, which are essential for delivering explicit guid-
 381 ance. As a result, the effectiveness of subsequent training stages is diminished to a certain extent.
 382

- 383 • *Removing Stage 2.* Building on Stage 1, Stage 2 focuses on optimizing the quality of generated
 384 global plans, thereby providing more effective high-level guidance for complex agent tasks. As
 385 indicated in Table 2 (1 → 3), eliminating Stage 2 results in a modest decline of 4.33% in perfor-
 386 mance relative to the model trained with all three stages.

- 387 • *Removing Stage 3.* Stage 3 aims to optimize the coordination between the global planner and
 388 executor, thereby enhancing the model’s end-to-end performance in agent tasks. As observed in
 389 Table 2 (1 → 2), excluding Stage 3 leads to a performance drop of 2.54%. Nevertheless, owing to
 390 the presence of fully implemented Stage 1 and Stage 2, the performance gap relative to the model
 391 trained through all three stages remains narrow and relatively small.

392 **Sequential Order of Stages.** We swap Stage 1 and Stage 2 to evaluate their influence on model
 393 performance. As seen in Table 2 (2 → 1 → 3), such reordering results in a slight performance
 394 decline of 1.58%. It supports the robustness of our original training sequence, which prioritizes the
 395 development of guidance-following capabilities before refining global plan generation skills. It is
 396 grounded in the need for a strong foundation of instruction follow-up to enhance the quality of global
 397 plans. Only with this foundation can the model make meaningful strides in developing its ability to
 398 generate global plans that effectively guide the action execution during agent task completion.
 399

400 4.2 FURTHER ANALYSIS

401 **AdaPlan vs. ReAct.** We compare the
 402 performance of the AdaPlan and Re-
 403 Act paradigms in agent tasks. Nei-
 404 ther of these paradigms undergoes ad-
 405 dditional training, with distinct prompt
 406 strategies employed to induce differ-
 407 ent thinking patterns in the model in-
 408 stead. As presented in Table 3, the
 409 experimental results indicate that our
 410 proposed AdaPlan exhibits greater ef-
 411 ficacy in enabling the model to accom-
 412 plish complex agent tasks by leverag-
 413 ing global planning as guidance, which
 414 outperforms ReAct by 12.76%.

415 **Unified Architecture vs. Iso- 416 lated Planner-Executor Architec- 417 ture.**

418 We conduct an evalua-
 419 tion against the isolated planner
 420 and executor framework (Erdogan
 421 et al., 2025) to validate the effec-
 422 tiveness of integrating both compo-
 423 nents within a unified model archi-
 424 tecture. In the isolated architecture
 425 setting, we employ the same back-
 426 bone model and separately train the
 427 planner and executor modules fol-
 428 lowing the Stage 1 and Stage 2 RL
 429 procedures described in PilotRL,
 430 with each component trained for 2
 431 epochs. As summarized in Table 4, the isolated architecture
 suffers from a performance drop of 5.63% compared to the unified architecture, in which both func-
 tionalities are learned jointly in an end-to-end manner, further emphasizing the importance of co-
 developing planning and execution capabilities within a single model.

Table 3: Analysis on the agent paradigms of *AdaPlan* and *ReAct* on In-Domain (ID) and Out-of-Domain (OOD) tasks. The best score of each model are in **bold**.

Backbone Model	Paradigm	ID	OOD	Avg.
Qwen2.5-7B -Instruct	AdaPlan	50.54	44.98	48.69
	ReAct	41.15	40.80	41.04
LLaMA3.1-8B -Instruct	AdaPlan	47.42	47.20	47.34
	ReAct	41.45	42.14	41.68
Qwen3-8B	AdaPlan	53.69	51.49	52.95
	ReAct	49.84	48.52	49.40

Table 4: Analysis of the *unified* and *isolated* planner-executor architectures on In-Domain (ID) and Out-of-Domain (OOD) tasks. The best scores are in **bold**.

Backbone Model	Architecture	ID	OOD	Avg.
Qwen2.5-7B -Instruct	Unified	72.93	59.75	68.53
	Isolated	68.94	55.18	64.36
LLaMA3.1-8B -Instruct	Unified	73.92	63.46	70.43
	Isolated	68.68	59.18	65.51
Qwen3-8B	Unified	74.18	60.95	69.77
	Isolated	72.66	56.02	67.11

8

432 **How planner, executor, and their coordination co-evolve during agent learning?** We
 433 analyze the evolution of reward scores for the
 434 global planner, the executor, and the end-to-
 435 end (E2E) performance in the training pro-
 436 cess of LLaMA3.1-8B-Instruct. As shown in
 437 Figure 3, the executor’s ability of plan adhe-
 438 sion saw a marked improvement during Stage 1
 439 and remained stable with slight growth in sub-
 440 sequent stages. The global planner’s per-
 441 formance, which generates high-level plans for ex-
 442 plicit guidance, exhibits a notable improvement
 443 in Stage 2 (epoch 2 & 3). It experiences a mild
 444 decline at the beginning of Stage 3, followed by a
 445 continuous upward trend. We speculate that this
 446 temporary drop reflects an adaptation period, dur-
 447 ing which the planner adjusts its generation to bet-
 448 ter align with the executor’s capabilities. Mean-
 449 while, the E2E reward increases steadily through-
 450 out the entire training process, indicating a con-
 451 sistent improvement in the system’s overall per-
 452 formance.

5 RELATED WORK

453 **LLM as Agent** The emergence of Large Language Models (LLMs) has driven research into the
 454 development of LLM-based agent systems (Zeng et al., 2024). The most common paradigm for
 455 LLM-based agent systems is ReAct (Yao et al., 2023), which integrates Chain-of-Thought (CoT)
 456 reasoning with agent actions in an interleaved manner to accomplish multiple tasks. However, this
 457 step-by-step reasoning framework struggles in scenarios demanding long-term strategic planning
 458 or complex multi-step coordination, e.g., household exploration (Shridhar et al., 2021) and games
 459 involving foresighted planning (Abdulhai et al., 2023), which highlights a pressing need for mech-
 460 anisms with coherent and long-term planning. Even though there have been efforts aimed to incor-
 461 porate explicit guidance into agent task completion (Deng et al., 2023; Zeng et al., 2024), the planner
 462 and executor are typically implemented in isolated architectural frameworks, leading to suboptimal
 463 instruction generation and execution alignment. Moreover, although closed-source proprietary mod-
 464 els often demonstrate strong performance in agent tasks, open-source models generally fall short in
 465 comparison (Liu et al., 2023). While recent studies have tried to collect expert trajectories from
 466 frontier LLMs (e.g., GPT-4) to fine-tune open-sourced models (Chen et al., 2023; 2024; Song et al.,
 467 2024; Zeng et al., 2024; Zhang et al., 2024b), such behavioral cloning strategy hinders the model’s
 468 generalization performance on out-of-distribution tasks. Therefore, it is necessary to introduce a
 469 more flexible training framework to cultivate models’ intrinsic generalization capabilities, e.g., re-
 470 inforcement learning (RL).

471 **Reinforcement Learning in LLMs** Compared to the supervised fine-tuning (SFT), reinforce-
 472 ment learning (RL) provides a more powerful paradigm for training LLM-based agents which are
 473 capable of autonomous planning, decision-making, and environmental interaction without explicit
 474 supervision (Guo et al., 2025; Jaech et al., 2024; Team et al., 2025). Among all the RL algorithms,
 475 GRPO (Shao et al., 2024; Guo et al., 2025) is specifically designed for LLMs, which has proven
 476 to be highly effective by replacing the traditional critic with a group-based evaluation strategy. Ef-
 477 forts have been made to enhance the agent capability in LLMs through the RL process, with notable
 478 works for information retrieval tasks (Jin et al., 2025; Song et al., 2025) and tool utilization scenar-
 479 os (Feng et al., 2025; Li et al., 2025). We situate our research on agent capability enhancement
 480 within the RL landscape for its effectiveness in fostering exploration and the emergence of novel
 481 strategies, and shift away from the commonly used ReAct framework (Yao et al., 2023), toward a
 482 global-plan-driven paradigm that supports more strategic and forward-looking decision-making.

6 CONCLUSION

483 In this paper, we introduce AdaPlan, an adaptive global plan-based agent paradigm. Based on the
 484 proposed paradigm, we put forward PilotRL, a global planning-guided training framework for LLM
 485 agents driven by progressive reinforcement learning. Experimental results indicate that PilotRL
 achieves excellent training outcomes in agent scenarios.

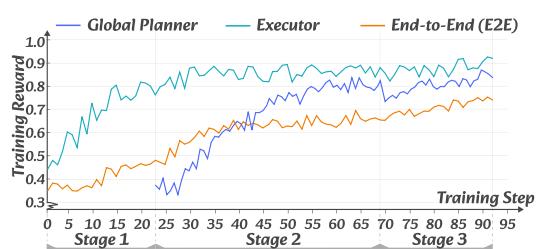


Figure 3: Normalized rewards for global plan-
 ner, executor and end-to-end (E2E) performance
 in training LLaMA3.1-8B-Instruct.

486

486 REFERENCES
487

488 Marwa Abdulhai, Isadora White, Charlie Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu,
489 and Sergey Levine. Lmrl gym: Benchmarks for multi-turn reinforcement learning with language
490 models. *arXiv preprint arXiv:2311.18232*, 2023.

491 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Ale-
492 man, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical
493 report. *arXiv preprint arXiv:2303.08774*, 2023.

494

495 Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao.
496 Fireact: Toward language agent fine-tuning. *arXiv preprint arXiv:2310.05915*, 2023.

497

498 Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and
499 Feng Zhao. Agent-flan: Designing data and methods of effective agent tuning for large language
500 models. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 9354–9366,
2024.

501

502 Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia,
503 Thien Huu Nguyen, and Yoshua Bengio. BabyAI: First steps towards grounded language learning
504 with a human in the loop. In *International Conference on Learning Representations*, 2019. URL
505 <https://openreview.net/forum?id=rJeXCo0cYX>.

506

507 Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V
508 Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation
509 model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

510

511 Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su.
512 Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing
Systems*, 36:28091–28114, 2023.

513

514 Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
515 Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models.
516 *arXiv preprint arXiv:2407.21783*, 2024.

517

518 Lutfi Eren Erdogan, Nicholas Lee, Sehoon Kim, Suhong Moon, Hiroki Furuta, Gopala Anu-
519 manchipalli, Kurt Keutzer, and Amir Gholami. Plan-and-act: Improving planning of agents for
520 long-horizon tasks. *arXiv preprint arXiv:2503.09572*, 2025.

521

522 Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang,
523 Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms.
524 *arXiv preprint arXiv:2504.11536*, 2025.

525

526 Dayuan Fu, Keqing He, Yejie Wang, Wentao Hong, Zhuoma GongQue, Weihao Zeng, Wei Wang,
527 Jingang Wang, Xunliang Cai, and Weiran Xu. Agentrefine: Enhancing agent generalization
528 through refinement tuning. In *The Thirteenth International Conference on Learning Representa-
tions*, 2025.

529

530 Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali
531 Farhadi. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE
conference on computer vision and pattern recognition*, pp. 4089–4098, 2018.

532

533 Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Ying-
534 han Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint
arXiv:2411.15594*, 2024.

535

536 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
537 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
538 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

539

Peter Hall. On kullback-leibler loss and density estimation. *The Annals of Statistics*, pp. 1491–1519,
1987.

540 Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Os-
 541 trow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint*
 542 *arXiv:2410.21276*, 2024.

543 Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec
 544 Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv*
 545 *preprint arXiv:2412.16720*, 2024.

546 Nicholas R Jennings, Katia Sycara, and Michael Wooldridge. A roadmap of agent research and
 547 development. *Autonomous agents and multi-agent systems*, 1:7–38, 1998.

548 Zixia Jia, Mengmeng Wang, Baichen Tong, Song-Chun Zhu, and Zilong Zheng. LangSuit-E:
 549 Planning, controlling and interacting with large language models in embodied text environ-
 550 ments. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Associa-
 551 tion for Computational Linguistics: ACL 2024*, pp. 14778–14814, Bangkok, Thailand, August
 552 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.879. URL
 553 <https://aclanthology.org/2024.findings-acl.879/>.

554 Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and
 555 Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement
 556 learning. *arXiv preprint arXiv:2503.09516*, 2025.

557 Haoran Li, Dadi Guo, Wei Fan, Mingshi Xu, Jie Huang, Fanpu Meng, and Yangqiu Song. Multi-step
 558 jailbreaking privacy attacks on chatgpt. *arXiv preprint arXiv:2304.05197*, 2023.

559 Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl. *arXiv preprint*
 560 *arXiv:2503.23383*, 2025.

561 Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao,
 562 Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint*
 563 *arXiv:2412.19437*, 2024.

564 Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding,
 565 Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint*
 566 *arXiv:2308.03688*, 2023.

567 Pattie Maes. Agents that reduce work and information overload. In *Readings in human-computer*
 568 *interaction*, pp. 811–821. Elsevier, 1995.

569 Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit
 570 Bansal, and Tushar Khot. ADaPT: As-needed decomposition and planning with language mod-
 571 els. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for*
 572 *Computational Linguistics: NAACL 2024*, pp. 4226–4252, Mexico City, Mexico, June 2024.
 573 Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.264. URL
 574 <https://aclanthology.org/2024.findings-naacl.264/>.

575 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
 576 Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical
 577 reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

578 Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng,
 579 Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings*
 580 *of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

581 Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi,
 582 Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions
 583 for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern*
 584 *recognition*, pp. 10740–10749, 2020.

585 Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew
 586 Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In
 587 *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=0IOX0YcCdTn>.

594 Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su.
 595 Llm-planner: Few-shot grounded planning for embodied agents with large language models. In
 596 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2998–3009, 2023.
 597

598 Huatong Song, Jinhao Jiang, Yingqian Min, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, Lei Fang,
 599 and Ji-Rong Wen. R1-searcher: Incentivizing the search capability in llms via reinforcement
 600 learning. *arXiv preprint arXiv:2503.05592*, 2025.

601 Yifan Song, Weimin Xiong, Xutian Zhao, Dawei Zhu, Wenhao Wu, Ke Wang, Cheng Li, Wei Peng,
 602 and Sujian Li. Agentbank: Towards generalized llm agents via fine-tuning on 50000+ interaction
 603 trajectories. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp.
 604 2124–2141, 2024.

605

606 Theodore Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas Griffiths. Cognitive architectures
 607 for language agents. *Transactions on Machine Learning Research*, 2023.

608

609 Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Fei Huang,
 610 and Yan Zhang. Zerosearch: Incentivize the search capability of llms without searching. *arXiv
 611 preprint arXiv:2505.04588*, 2025.

612 Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu,
 613 Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly
 614 capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

615

616 Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun
 617 Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with
 618 llms. *arXiv preprint arXiv:2501.12599*, 2025.

619 Michael Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *The knowl-
 620 edge engineering review*, 10(2):115–152, 1995.

621

622 Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Dingwen
 623 Yang, Chenyang Liao, Xin Guo, Wei He, et al. Agentgym: Evolving large language model-based
 624 agents across diverse environments. *arXiv preprint arXiv:2406.04151*, 2024.

625

626 Weimin Xiong, Yifan Song, Qingxiu Dong, Bingchan Zhao, Feifan Song, Xun Wang, and Sujian Li.
 627 Mpo: Boosting llm agents with meta plan optimization. *arXiv preprint arXiv:2503.02682*, 2025.

628

629 An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li,
 630 Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint
 631 arXiv:2412.15115*, 2024.

632

633 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu,
 634 Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint
 635 arXiv:2505.09388*, 2025.

636

637 Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao.
 638 React: Synergizing reasoning and acting in language models. In *International Conference on
 639 Learning Representations (ICLR)*, 2023.

640

641 Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Pinjia He, Shuming Shi, and
 642 Zhaopeng Tu. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher. *arXiv preprint
 643 arXiv:2308.06463*, 2023.

644

645 Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agent-
 646 tuning: Enabling generalized agent abilities for llms. In *Findings of the Association for Compu-
 647 tational Linguistics ACL 2024*, pp. 3053–3077, 2024.

648

649 Jianguo Zhang, Tian Lan, Rithesh Murthy, Zhiwei Liu, Weiran Yao, Ming Zhu, Juntao Tan, Thai
 650 Hoang, Zuxin Liu, Liangwei Yang, et al. Agentohana: Design unified data and training pipeline
 651 for effective agent learning. *arXiv preprint arXiv:2402.15506*, 2024a.

648 Jianguo Zhang, Tian Lan, RN Rithesh, Zhiwei Liu, Weiran Yao, Juntao Tan, Thai Quoc Hoang,
649 Liangwei Yang, Yihao Feng, Zuxin Liu, et al. The agent ohana: Designing unified data and
650 training pipeline for effective agent learning. In *ICLR 2024 Workshop on Large Language Model*
651 (*LLM*) *Agents*, 2024b.

652 Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang,
653 Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and
654 chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

655 Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei
656 Liu. Deepresearcher: Scaling deep research via reinforcement learning in real-world environ-
657 ments. *arXiv preprint arXiv:2504.03160*, 2025.

658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701

APPENDIX

A Group Relative Policy Optimization (GRPO) 14B Experiment Details 14

B.1 Datasets	14
B.2 Baselines	16
B.3 Ablation Study and Further Analysis Details	16
B.4 Environment and Hardware Configurations	18

C Prompts 18D Case Studies 32E Ethics Statement 32

A 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 $\pi_{\theta_{\text{ref}}}$:

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

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.

B EXPERIMENT DETAILS

B.1 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 5.

- **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

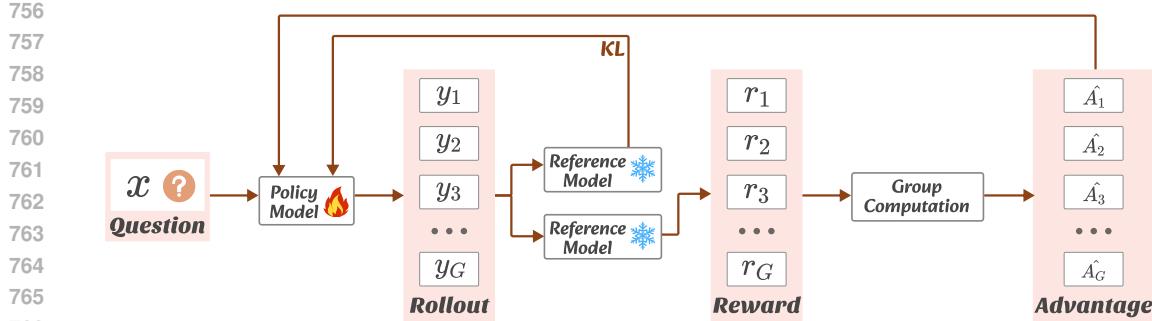


Figure 4: An illustration for the Group Relative Policy Optimization (GRPO) pipeline.

Table 5: Statistics of datasets for training and evaluation.

Classification	Dataset	#Training Num.	#Testing Num.
<i>In-Domain</i>	ALFWorld	3000	321
	IQA	1465	162
	TextCraft	400	74
	Wordle	860	95
<i>Out-of-Domain</i>	BabyAI	–	400
	MAZE	–	215

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.

810
811 B.2 BASELINES812
813 In this section, we provide a comprehensive overview of the various methods that serve as baselines
in our comparison.

- 814 • **Close-Sourced Models:** Closed-source models are considered to represent the current
815 state-of-the-art in LLM capabilities and are regarded as the most competitive baseline meth-
816 ods. We have selected GPT-4o and GPT-4o-mini (Hurst et al., 2024) to assess the upper
817 bound of the model performance on agent tasks.
- 818 • **Open-Sourced Agent-Specific Models:** These models refer to models that were trained
819 specifically on agent-task datasets. We have selected Agent-FLAN-7B (Chen et al., 2024),
820 LLaMA-xLAM-2-8B-fc-r (Zhang et al., 2024a) and DeepResearcher-7B (Zheng et al.,
821 2025) to represent the open-sourced agent-specific models for comparison to assess Pi-
822 lotRL’s relative advantages. Specifically, the backbone model of DeepResearcher-7B is
823 Qwen2.5-7B-Instruct (Yang et al., 2024), which facilitates a more direct comparison with
824 Qwen2.5-7B-Instruct + PilotRL.
- 825 • **Naive Response:** It refers to the case where the model directly generates responses without
826 any training (e.g., SFT, RL, etc.) or prompting (e.g., ReAct) strategies.
- 827 • **ReAct** (Yao et al., 2023): It is the prompting strategy that integrates single-step reasoning
828 with the execution of the current action, which is a common agent paradigm.
- 829 • **MPO** (Xiong et al., 2025): The Meta Plan Optimization (MPO) framework improves the
830 agent’s planning capabilities by integrating explicit guidance into the decision-making pro-
831 cess. As an external plug-and-play planner, MPO provides the model with high-level meta-
832 plans that serve as structured guidance during task execution. One key distinction between
833 MPO and PilotRL lies in the integration and training of the planner and executor compo-
834 nents. In our approach, both the planner and executor reside within the same model and are
835 trained jointly. In contrast, MPO maintains separate models for planning and execution,
836 where only the planner is trained while the executor’s parameters remain frozen, leading to
837 limited coordination between the two components.
- 838 • **Supervised Fine-Tuning (SFT):** This training strategy is widely adopted in a series of
839 studies (Chen et al., 2024; Song et al., 2024; Xi et al., 2024; Zeng et al., 2024; Zhang et al.,
840 2024b; Fu et al., 2025). However, existing studies have shown that compared to RL, SFT
841 generally exhibits weaker generalization capabilities on new tasks—particularly when the
842 training data consists of multi-step trajectories for problem-solving (Shao et al., 2024; Team
843 et al., 2025). This is because such trajectories may contain redundant or suboptimal paths to
844 task completion. Moreover, SFT tends to bias the model toward previously seen execution
845 paths, limiting its ability to adapt or generalize to novel scenarios through compositional or
846 analogical reasoning. During SFT, we use the same datasets with PilotRL. In addition, we
847 incorporate the original agent-environment interaction trajectories into training, a setting
848 that differs from **Vanilla RL** and our **PilotRL**. Furthermore, we generate global plans for
849 guiding task completion using DeepSeek-V3, and feed both the interaction trajectories and
850 the corresponding global plans into the model during training. This setup allows us to
851 compare PilotRL over existing baselines under a more fair and controlled experimental
852 condition.
- 853 • **Vanilla RL:** We also conduct training with the naive reinforcement learning process utiliz-
854 ing the Group Relative Policy Optimization (GRPO) (Shao et al., 2024) algorithm. Here
855 we employ only the format and end-to-end (E2E) performance as the reward metrics. This
856 baseline is for validating the effectiveness of adaptive global planning.

857 B.3 ABLATION STUDY AND FURTHER ANALYSIS DETAILS

859
860 In this section, we report the original performance scores of the models on each benchmark during
861 training stage and training sequence ablation, as well as reward design ablation, as depicted in
862 Table 6, Table 7 and Table 8.863 **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-

864
 865 **Table 6: Original scores for each benchmark of the ablation study on multiple training stages**
 866 **and sequential order.** It is the detailed version of [Table 2](#). “Order” is the sequential order of
 867 Stage 1, 2, and 3 during training. Specifically, “1 & 2 & 3” refers to a joint training configuration
 868 in which reward functions from all three stages are merged and optimized concurrently, where the
 869 target model generates global plans independently throughout the entire training process. The best
 870 and second best scores of each model are in **bold** and underlined.
 871

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	<u>75.37</u>	77.69	61.56	57.93	68.53
	LLaMA3.1-8B-Instruct	78.53	<u>72.78</u>	64.76	<u>79.61</u>	68.24	58.68	70.43
	Qwen3-8B	<u>72.51</u>	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	<u>55.81</u>	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	<u>72.43</u>	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	<u>72.61</u>	<u>70.59</u>	83.27	64.73	53.28	<u>69.44</u>
Sequential Order of Stages								
2 → 1 → 3	Qwen2.5-7B-Instruct	<u>70.12</u>	66.08	73.98	77.85	59.63	55.67	67.22
	LLaMA3.1-8B-Instruct	<u>77.25</u>	73.15	<u>64.02</u>	<u>77.63</u>	65.98	58.14	<u>69.36</u>
	Qwen3-8B	72.94	73.86	68.55	78.02	<u>65.07</u>	54.80	68.87

895
 896
 897 **Table 7: Original scores for each benchmark of the agent paradigm analysis.** It is the detailed
 898 version of [Table 3](#). The best scores of each model are in **bold**. It shows that *AdaPlan* consistently
 899 outperforms *ReAct* on both in-domain and out-of-domain agent tasks across all models, demon-
 900 strating performance gains of 18.64%, 13.58%, 7.19% on Qwen2.5-7B-Instruct, LLaMA3.1-8B-Instruct,
 901 and Qwen3-8B, respectively.

Backbone Model	Paradigm	ALFWORLD	IQA	TextCraft	Wordle	BabyAI	MAZE
		In-Domain (ID)				Out-of-Domain (OOD)	
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%)

902
 903 Instruct + PilotRL, we normalized all reward scores to the range [0, 1] for visualization and compar-
 904 ison purposes. The reward metrics include:

905
 906
 907

- **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.

918
 919 **Table 8: Original scores for each benchmark of the planner-executor architecture analysis.**
 920 It is the detailed version of Table 4. The best scores of each model are in **bold**. It shows that the
 921 *unified architecture* consistently outperforms *isolated architecture* on both in-domain and out-of-
 922 domain agent tasks across all models, with measured improvements of 6.48%, 7.51%, 3.96% on
 923 Qwen2.5-7B-Instruct, LLaMA3.1-8B-Instruct, and Qwen3-8B, respectively.

Backbone Model	Architecture	ALFWorld		IQA <i>In-Domain (ID)</i>		TextCraft		Wordle		BabyAI <i>Out-of-Domain (OOD)</i>	
		Isolated	Unified	68.85	64.18	72.60	70.14	58.29	52.07	61.56 (↑ 5.61%)	57.93 (↑ 11.25%)
Qwen2.5-7B-Instruct	Isolated	70.80 (↑ 2.83%)	67.84 (↑ 5.70%)	75.37 (↑ 3.82%)	77.69 (↑ 10.76%)	71.05	62.71	55.64	51.49	68.24 (↑ 8.82%)	58.68 (↑ 5.46%)
LLaMA3.1-8B-Instruct	Isolated	71.87	70.83	60.96	79.61 (↑ 12.05%)	82.23	60.55	51.49	56.28 (↑ 7.81%)	56.62 (↑ 9.96%)	
Qwen3-8B	Isolated	71.74	67.71	68.96	83.65 (↑ 1.73%)						
	Unified	72.51 (↑ 1.07%)	69.06 (↑ 1.99%)	71.48 (↑ 3.65%)							

931
 932 • **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
 933 for assessing overall system behavior.

936 B.4 ENVIRONMENT AND HARDWARE CONFIGURATIONS

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

942 C PROMPTS

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

Prompt: Global Plan Generation - ALFWorld

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

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

- 956 1. go to recep
- 957 2. task obj from recep
- 958 3. put obj in/on recep
- 959 4. open recep
- 960 5. close recep
- 961 6. toggle obj recep
- 962 7. clean obj with recep
- 963 8. heat obj with recep
- 964 9. cool obj with recep

965 where obj and recep correspond to objects and receptacles.

966 # Task
 967 {task}

```

972
973 # Previous Global Plan
974 {global_plan} [optional]
975
976 # Execution Step Index
977 {execution_step_index}
978
979 # Accumulated Observation
980 {observation} [optional]
981
982 -----
983
984 Output Format:
985     ````json
986     [
987     Step 1: ...
988     Step 2: ...
989     ...
990     ', ...
991
992
993
994

```

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.

```

1010
1011 # Task
1012 {task}
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

```

Previous Global Plan
{global_plan} [optional]

Execution Step Index
{execution_step_index}

Accumulated Observation
{observation} [optional]

Output Format:

```

1026
1027     '''json
1028     [
1029     Step 1: ...
1030     Step 2: ...
1031     ...
1032     ', ...]
1033
1034

```

Prompt: Global Plan Generation - TextCraft

1035 You are given a few useful crafting recipes to craft items in Minecraft. Craft command can
 1036 be understood as follows: craft [target] using [ingredients], where target is item/object gen-
 1037 erated by the craft command as output and ingredient are the inputs. You are given an agent
 1038 that can “craft” or “fetch” objects. You can take the help of crafting commands below to
 1039 create new objects. Based on the task description, the previous global plan, and accumulated
 1040 observation of agent interactions with the environment, please generate all possible step-by-
 1041 step global plans, which serve as high-level, natural guidance to assist in planning. Maintain
 1042 the plan for all steps preceding the execution step index, while selectively modifying the plan
 1043 for steps following the execution step index. Each global plan can use at most ONE of the
 1044 provided crafting commands.
 1045

```

1046
1047 # Task
1048 {task}
1049

```

```

1050
1051 # Previous Global Plan
1052 {global_plan} [optional]
1053

```

```

1054 # Execution Step Index
1055 {execution_step_index}
1056

```

```

1057 # Accumulated Observation
1058 {observation} [optional]
1059
-----
```

Output Format:

```

1060
1061     '''json
1062     [
1063     Step 1: ...
1064     Step 2: ...
1065     ...
1066     ', ...]
1067
1068
1069

```

Prompt: Global Plan Generation - Wordle

1070 You are an expert wordle player. Based on the task description, the previous global plan,
 1071 and accumulated observation of agent interactions with the environment, please generate all
 1072 possible step-by-step global plans for the wordle task, which serve as high-level, natural
 1073 guidance to assist in planning. Maintain the plan for all steps preceding the execution step
 1074 index, while selectively modifying the plan for steps following the execution step index.
 1075 Your objective is to guess a hidden 5 letter word. You have 6 attempts to guess it correctly
 1076 and you should try to guess it in as few attempts as possible. When guessing the word, you
 1077 should format your word as a space separated sequence of letters, like “s h i r e” for example.
 1078

1080
 1081 After guessing the word, you will receive feedback from the game environment in the form
 1082 of a sequence of 5 space separated letters like “b y g g b”, where each letter indicates some
 1083 information about the hidden word. The environment will return one of three letters - “b”,
 1084 “g”, or “y” – for each letter in the word you guessed. Here is the meaning of each letter:
 1085 • “b”: If the environment returns a “b”, it means that the letter at that position in your
 1086 guessed word is not in the hidden word.
 1087 • “y”: If the environment returns a “y”, it means that the letter at that position in your
 1088 guessed word is in the hidden word but is not in the correct position.
 1089 • “g”: If the environment returns a “g”, it means that the letter at that position in your
 1090 guessed word is in the hidden word and is in the correct position.
 1091
 1092
 1093 **# Task**
 1094 {task}
 1095
 1096 **# Previous Global Plan**
 1097 {global_plan} [optional]
 1098
 1099 **# Execution Step Index**
 1100 {execution_step_index}
 1101
 1102 **# Accumulated Observation**
 1103 {observation} [optional]
 1104
 1105 -----
 1106
 1107 **Output Format:**
 1108 ```json
 1109 [
 1110 Step 1: ...
 1111 Step 2: ...
 1112 ...
 1113 ", ...]
 1114
 1115

Prompt: Global Plan Generation - BabyAI

1116 You are an exploration master that wants to finish every goal you are given. You are placed
 1117 in a room and you need to accomplish the given goal with actions. Based on the task de-
 1118 scription, the previous global plan, and accumulated observation of agent interactions with
 1119 the environment, please generate all possible step-by-step global plans, which serve as high-
 1120 level, natural guidance to assist in planning. Maintain the plan for all steps preceding the
 1121 execution step index, while selectively modifying the plan for steps following the execution
 1122 step index.
 1123

1124
 1125 The action list you can take:
 1126

- 1127 1. turn right
- 1128 2. turn left
- 1129 3. move forward
- 1130 4. go to <obj> <id>
- 1131 5. pick up <obj> <id>
- 1132 6. go through <door> <id>; <door> must be an open door.

```

1134
1135    7. toggle and go through <door> <id>: <door> can be a closed door or a locked
1136    door. If you want to open a locked door, you need to carry a key that is of the same
1137    color as the locked door.
1138    8. toggle: there is a closed or locked door right in front of you and you can toggle it.
1139    where <obj> and <id> correspond to objects and index number.
1140
1141 # Task
1142 {task}
1143
1144 # Previous Global Plan
1145 {global_plan} [optional]
1146
1147 # Execution Step Index
1148 {execution_step_index}
1149
1150 # Accumulated Observation
1151 {observation} [optional]
1152
1153 -----
1154
1155 Output Format:
1156
1157     '''json
1158     [
1159     Step 1: ...
1160     Step 2: ...
1161     ...
1162     ', ...
1163     ''
1164
1165 Prompt: Global Plan Generation - MAZE
1166
1167 You are an expert maze solver. Your objective is to reach the goal in as few steps as possible.
1168 Based on the task description, the previous global plan, and accumulated observation
1169 of agent interactions with the environment, please generate all possible step-by-step global
1170 plans, which serve as high-level, natural guidance to assist in planning. Maintain the plan for
1171 all steps preceding the execution step index, while selectively modifying the plan for steps
1172 following the execution step index. Your objective is to reach the goal in as few steps as
1173 possible. When you move right, you increase your y position by 1. When you move down,
1174 you increase your x position by 1.
1175
1176 The action list you can take:
1177
1178 1. move up
1179 2. move down
1180 3. move left
1181 4. move right
1182
1183 -----
1184
1185 For instance, given the current environment state: The goal is at position 8, 6. Your current
1186 position is at position 1, 1. There are walls to your left, above you, below you. The index of
1187 already executed steps is 0. The possible global plans could be:

```

```

1188
1189 [“
1190 Step 1: move right (from 1, 1 to 1, 2)
1191 Step 2: move right (from 1, 2 to 1, 3)
1192 Step 3: move right (from 1, 3 to 1, 4)
1193 Step 4: move down (from 1, 4 to 2, 4)
1194 Step 5: move down (from 2, 4 to 3, 4)
1195 Step 6: move down (from 3, 4 to 4, 4)
1196 Step 7: move down (from 4, 4 to 5, 4)
1197 Step 8: move down (from 5, 4 to 6, 4)
1198 Step 9: move down (from 6, 4 to 7, 4)
1199 Step 10: move down (from 7, 4 to 8, 4)
1200 Step 11: move right (from 8, 4 to 8, 5)
1201 Step 12: move right (from 8, 5 to 8, 6)
1202 ”, ...]
1203
1204
1205 # Task
1206 {task}
1207
1208 # Previous Global Plan
1209 {global_plan} [optional]
1210
1211 # Execution Step Index
1212 {execution_step_index}
1213
1214 # Accumulated Observation
1215 {observation} [optional]
1216
1217
1218
1219 Output Format:
1220
1221 ````json
1222 [
1223 Step 1: ...
1224 Step 2: ...
1225 ...
1226 ”, ...]
1227
1228
1229 Prompt: Global Plan Selection (for the generate-then-select strategy)
1230
1231 You are given several global plans serving as high-level, natural guidance to assist in
1232 planning. Based on the task description, accumulated observation of agent interactions with
1233 the environment, and the current index of execution step, please select the most suitable
1234 global plan from all available global plans for task completion.
1235
1236 When you select the global plan, consider evaluating the following aspects to identify the
1237 optimal choice based on comprehensive criteria:
1238
1. Correctness: Does the global plan correctly and accurately address the task require-
1239 ments?
1240
2. Executability: Is the global plan clearly structured, easy to interpret, and are the
1241 individual steps logically sound and actionable?

```

```

1242
1243 3. Standardization: Does the global plan adhere to a consistent and standardized for-
1244 mat?
1245
1246 # Task
1247 {task}
1248
1249 # Available Global Plans
1250 {global_plans}
1251
1252 # Execution Step Index
1253 {execution_step_index}
1254
1255 # Accumulated Observation
1256 {observation} [optional]
1257

```

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.

```

1273 # Task
1274 {task}
1275
1276 # Global Plan
1277 {global_plan}
1278
1279 # Execution Step Index
1280 {execution_step_index}
1281
1282 # Accumulated Observation
1283 {observation} [optional]
1284
1285
1286

```

Output Format:

```

1288 ```.json
1289 {
1290     "correctness_score": xxx,
1291     "correctness_reason": "...",
1292     "executability_score": xxx,
1293     "executability_reason": "...",
1294     "standardization_score": xxx,
1295     "standardization_reason": "..."

```

1296
 1297 }
 1298 '.'
 1299

Prompt: Environmental Feedback

1300 Based on the task description and the reference agent-environment interaction in which the
 1301 agent has finally accomplished the task, please generate the environmental feedback for the
 1302 agent's action and determine whether the current action has reached the final goal. If the
 1303 agent's action has reached the final goal, please output "Task Completed!"; else, the feedback
 1304 should be in the following format: "Observation: ...".

Task

{task}

Reference Interaction

{ref_interaction}

Previous Observation

{observation} [optional]

Agent Action

{agent_action}

Prompt: Execution Generation - ALFWorld

1320
 1321 Interact with a household to solve a task. Imagine you are an intelligent agent in a household
 1322 environment and your target is to perform actions to complete the task goal. At the beginning
 1323 of your interactions, you will be given the detailed description of the current environment and
 1324 your goal to accomplish. For each of your turn, you will be given the observation of the last
 1325 turn. You should choose from two actions: "Thought" or "Action". If you choose "Thought",
 1326 you should first think about the current condition and plan for your future actions, and then
 1327 output your action in this turn. Your output must strictly follow this format: "Thought: your
 1328 thoughts. Action: your next action"; If you choose "Action", you should directly output the
 1329 action in this turn. Your output must strictly follow this format: "Action: your next action".
 1330

1331 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 recep

1332 where obj and recep correspond to objects and receptacles.

1333
 1334
 1335
 1336
 1337
 1338
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349
 Reminder:

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

1350
 1351 2. Think when necessary, but prioritize direct action wherever possible throughout the
 1352 process.
 1353
 1354 **# Example**
 1355 {example}
 1356
 1357 **# Task**
 1358 {task}
 1359
 1360 **# Global Plan**
 1361 {global_plan}
 1362
 1363 **# Previous Observation**
 1364 {observation} [optional]
 1365
 1366
 1367

Prompt: Execution Generation - IQA

1368
 1369
 1370 Imagine you are an intelligent agent in a dynamic visual environment and your target is to
 1371 perform actions to complete the task goal. At the beginning of your interactions, you will be
 1372 given the detailed description of the current environment and your goal to accomplish. For
 1373 each of your turn, you will be given the observation of the last turn. You should choose from
 1374 two actions: “Thought” or “Action”. If you choose “Thought”, you should first think about
 1375 the current condition and plan for your future actions, and then output your action in this
 1376 turn. Your output must strictly follow this format: “Thought: your thoughts. Action: your
 1377 next action”; If you choose “Action”, you should directly output the action in this turn. Your
 1378 output must strictly follow this format: “Action: your next action”.
 1379

1380 The action list you can take:
 1381 1. move ahead
 1382 2. turn left
 1383 3. turn right
 1384 4. open obj
 1385 5. answer [True]/[False]
 1386
 1387 where obj correspond to objects.
 1388
 1389 Reminder:
 1390 1. The action is restricted to those listed as available. Actions not included in the
 1391 provided list are considered invalid.
 1392 2. Think when necessary, but prioritize direct action wherever possible throughout the
 1393 process.
 1394
 1395 **# Example**
 1396 {example}
 1397
 1398 **# Task**
 1399 {task}
 1400
 1401 **# Global Plan**
 1402 {global_plan}
 1403

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

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.

1458
1459
1460

- “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.

1461 For each of your turn, you will be given the observation of the last turn. You should choose
1462 from two actions: “Thought” or “Action”. If you choose “Thought”, you should first think
1463 about the current condition and plan for your future actions, and then output your action in
1464 this turn. Your output must strictly follow this format: “Thought: your thoughts. Action:
1465 your next action”; If you choose “Action”, you should directly output the action in this turn.
1466 Your output must strictly follow this format: “Action: your next action”.

1467
1468

Reminder:

1469
1470
1471
1472
1473

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.

1474
1475
1476**# Example**

{example}

1477
1478
1479**# Task**

{task}

1480
1481
1482**# Global Plan**

{global_plan}

1483
1484
1485**# Previous Observation**

{observation} [optional]

1486
1487**Prompt: Execution Generation - BabyAI**1489
1490
1491
1492
1493
1494
1495
1496

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”.

1497
1498

The action list you can take:

1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511

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.

1512

1513

1514

1515

1516

1517

1518

1519

1520

1521

1522

1523

1524

1525

1526

1527

1528

1529

1530

1531

1532

1533

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}

1566
 1567 **# Previous Observation**
 1568 {observation} [optional]
 1569

1570 **Prompt: Adherence Degree Judgment (for Equation (3))**
 1571

1572 You are an expert in agent tasks. You are tasked with evaluating the agent's execution of
 1573 a given global plan. Specifically, you are to assess the degree of compliance between the
 1574 agent's actions and the strategic guidance outlined in the global plan. Rate it from 0 to 2
 1575 points, and explain the reason.

1576
 1577 **2 Point Answer Criteria:**

1578 The agent's execution strictly adheres to the guidance provided in the global plan. All
 1579 actions are logically aligned with the plan's objectives and are carried out as instructed.

1580
 1581 **1 Point Answer Criteria:**

1582 The agent's execution demonstrates a partial alignment with the global plan, allowing for
 1583 minor deviations. For example, in cases where the plan suggests the use of multiple tools,
 1584 the agent may use at least one relevant tool to support the execution, as long as it does not
 1585 contradict the overall guidance.

1586
 1587 **0 Point Answer Criteria:**

1588 The agent's execution departs or contradicts the global plan, or contains garbled characters,
 1589 format errors, disorder, and irrelevant information.

1590 **# Task**

1591 {task}

1592
 1593 **# Global Plan**

1594 {global_plan}

1595
 1596 **# Execution Step Index**

1597 {execution_step_index}

1598
 1599 **# Agent Action**

1600 {agent_action}

1601
 1602 **Output Format:**

1603
 1604 ```json
 1605 {
 1606 "score": xxx,
 1607 "reason":
 1608 }
 1609
 1610

1611 **Prompt: E2E Performance Judgment (for Equation (4))**
 1612

1613 You are an expert in agent tasks. Please evaluate the end-to-end (E2E) performance of the
 1614 agent during its interaction with a given environment. The goal is to assess whether the
 1615 agent accomplishes the target task efficiently and directly, without unnecessary detours or
 1616 redundancies. Rate it from 0 to 2 points, and explain the reason.

1617
 1618 **2 Point Answer Criteria:**

1619 1. The agent successfully completes the task in a direct and efficient manner.

1620
 1621 2. There are no unnecessary steps or redundant actions in the interaction trajectory.
 1622 ***1 Point Answer Criteria:***
 1623 1. The task is ultimately completed, but the process includes some level of redundancy
 1624 or unintended topic drift.
 1625 2. While the final objective is met, there may be deviations from the optimal path.
 1626 ***0 Point Answer Criteria:***
 1627 1. The agent fails to achieve the final task objective.
 1628 2. Contains significant deviations, errors, or inability to progress towards the goal.
 1629
 1630
 1631
 1632 **# Task**
 1633 {task}
 1634
 1635 **# Agent-Environment Interaction**
 1636 {accumulated_context}
 1637
 1638 **# Reference Interaction**
 1639 {ref_interaction}

1640
 1641
 1642 **Output Format:**
 1643 ```json
 1644 {
 1645 "score": xxx,
 1646 "reason": "..."
 1647 }
 1648
 1649

Prompt: E2E Performance Evaluation (*LLM-as-Judge*)

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, focusing on two key dimensions:

- **Task Success:** Did the agent achieve the final goal?
- **Interaction Efficiency:** Was the path direct, logical, and free of redundancy or detours?

Assign a score from 0 to 100 and provide a clear justification. Please use the following criteria, and explain the reason.

90–100: Highly Successful and Efficient.

The agent demonstrates near-optimal behavior. All of the following must hold:

1. The final task objective is fully and correctly completed.
2. The interaction trajectory is direct and logically structured.
3. There are no redundant, repetitive, or off-topic actions.
4. Any minor errors (e.g., phrasing) do not impede progress.

70–89: Successful but Inefficient.

The task is completed, but with non-critical inefficiencies. At least one of the following applies:

1. The agent takes unnecessary steps or detours before completing the task.
2. There are minor errors or invalid actions that require recovery.
3. Brief topic drift or redundant reasoning occurs but is self-corrected.
4. The overall strategy works but is suboptimal in efficiency.

1674
 1675 **50–69: Partially Successful.**
 1676 Significant progress is made, but the task is not fully completed or requires excessive effort.
 1677 At least one of the following applies:
 1678 1. The agent fails to reach the final goal, but completes most subtasks.
 1679 2. Completion requires major detours, repeated failures, or external hints.
 1680 3. The agent shows inconsistent planning or weak strategy.
 1681 4. Some actions are irrelevant or counterproductive, though not fatal.
 1682
 1683 **0–49: Failure or Severely Flawed.**
 1684 The agent fails to make meaningful progress. At least one of the following applies:
 1685 1. The agent fails to complete the core objective.
 1686 2. There is complete task misunderstanding or topic deviation.
 1687 3. The interaction contains repeated invalid actions or infinite loops.
 1688 4. The agent shows no coherent strategy and cannot recover from errors.
 1689
 1690
 1691 **# Task**
 1692 {task}
 1693
 1694 **# Agent-Environment Interaction**
 1695 {accumulated_context}
 1696
 1697 **# Reference Interaction**
 1698 {ref_interaction}
 1699
 1700
 1701 **Output Format:**
 1702 ```json
 1703 {
 1704 "score": xxx,
 1705 "reason": "..."
 1706 }
 1707 '''

D CASE STUDIES

1710
 1711
 1712 For agent tasks involving multi-step decision-making, generating a global plan to guide the execu-
 1713 tion of each step is crucial. This is because models may forget the previous context after executing
 1714 multiple steps, leading to redundant actions or failure to accomplish the task. As shown in Figure 5,
 1715 the red annotations indicate redundant interaction trajectories during the problem-solving process.
 1716 When the agent has already moved three steps to the right, it forgets that the red ball should be
 1717 directly on its left and continues to move forward, resulting in a large amount of redundant execu-
 1718 tions. In contrast, as depicted in Figure 6, with the guidance of a global plan, the agent can clearly
 1719 recognize its relative position of the target, thereby efficiently completing the task.
 1720

E ETHICS STATEMENT

1721
 1722
 1723 The experimental design in our paper was carefully planned to ensure that all data used for train-
 1724 ing and evaluation were obtained through legitimate means and adhered to relevant privacy laws
 1725 and regulations. We have also provided detailed descriptions of our methodologies, algorithms, and
 1726 prompts to enable reproducibility.
 1727

1728
1729
1730
1731
1732
1733

1734 **Task** 
1735 **Instruction**

1736 You are placed in a room and you need to accomplish the given goal with actions. Your goal
1737 is to: go to the red ball. In front of you in this room, you can see several objects: There is a
1738 red ball 1 4 steps in front of you and 3 steps to your right. The room has walls around you.
1739 You are facing a wall 6 steps away. You are not carrying anything.



1739 **React**



1740 **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
1741 reach it. **Action:** turn right

1742 **Observation:** In front of you in this room, you can see several objects: The room has walls around you. You
1743 are facing a wall 5 steps away. You are not carrying anything.

1744 **Thought:** I need to keep moving forward to find the red ball. **Action:** move forward

1745 **Observation:** In front of you in this room, you can see several objects: The room has walls around you. You
1746 are facing a wall 4 steps away. You are not carrying anything.

1747 **Thought:** I still need to move forward to get closer to the red ball. **Action:** move forward

1748 **Observation:** In front of you in this room, you can see several objects: The room has walls around you. You
1749 are facing a wall 3 steps away. You are not carrying anything.

1750 **Thought:** I need to continue moving forward to get closer to the red ball. **Action:** move forward

1751 **Observation:** In front of you in this room, you can see several objects: The room has walls around you. You
1752 are facing a wall 2 steps away. You are not carrying anything.

1753 **Thought:** The red ball should be nearby, I should be getting closer. **Action:** move forward

1754 **Observation:** In front of you in this room, you can see several objects: The room has walls around you. You
1755 are facing a wall 1 steps away. You are not carrying anything.

1756 **Thought:** I can't move forward anymore. I should turn right to find the red ball. **Action:** turn right

1757 **Observation:** In front of you in this room, you can see several objects: The room has walls around you. You
1758 are facing a wall 1 steps away. You are not carrying anything.

1759 **Thought:** I need to turn right again to find the red ball. **Action:** turn right

1760 **Observation:** In front of you in this room, you can see several objects: The room has walls around you. You
1761 are facing a wall 6 steps away. You are not carrying anything.

1762 **Thought:** I need to turn left to move towards the red ball. **Action:** turn left

1763 **Observation:** In front of you in this room, you can see several objects: The room has walls around you. You
1764 are facing a wall 1 steps away. You are not carrying anything.

1765 **Thought:** The red ball should be close by, I should turn left to find it. **Action:** turn left

1766 **Observation:** In front of you in this room, you can see several objects: The room has walls around you. You
1767 are facing a wall 1 steps away. You are not carrying anything.

1768 **Thought:** I need to turn left again to keep searching for the red ball. **Action:** turn left

1769 **Observation:** In front of you in this room, you can see several objects: There is a red ball 1 4 steps in front
1770 of you and 1 steps to your left. The room has walls around you. You are facing a wall 6 steps away. You
1771 are not carrying anything.

1772 **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
1773 reach it. **Action:** turn left

1774 **Observation:** In front of you in this room, you can see several objects: The room has walls around you. You
1775 are facing a wall 6 steps away. You are not carrying anything.

1776 **Thought:** I need to keep moving forward to find the red ball. **Action:** move forward

1777 **Observation:** In front of you in this room, you can see several objects: The room has walls around you. You
1778 are facing a wall 5 steps away. You are not carrying anything.

1779 **Thought:** I finally found the red ball! I need to go directly to the red ball. **Action:** go to red ball 1

1780
1781

1777 Figure 5: Case study of ReAct (Yao et al., 2023) on BabyAI (Chevalier-Boisvert et al., 2019).
1778
1779
1780
1781

1782
 1783
 1784
 1785
 1786
 1787
 1788
 1789
 1790
 1791
 1792
 1793
 1794
 1795
 1796
 1797
 1798
 1799
 1800
 1801
 1802
 1803
 1804
 1805
 1806
 1807
 1808
 1809
 1810
 1811
 1812
 1813
 1814
 1815
 1816
 1817
 1818
 1819
 1820
 1821
 1822
 1823
 1824
 1825
 1826
 1827
 1828
 1829
 1830
 1831
 1832
 1833
 1834
 1835

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.

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