

# The Interplay of Harness Design and Post-Training in LLM Agents

Kyungmin Kim<sup>1,\*</sup>, Youngbin Choi<sup>1,\*</sup>, Seoyeon Lee<sup>1</sup>, Suhyeon Jun<sup>2</sup>,  
Dongwoo Kim<sup>1,2,†</sup>, Sangdon Park<sup>1,2,†</sup>

<sup>1</sup>Graduate School of Artificial Intelligence, POSTECH,

<sup>2</sup>Department of Computer Science and Engineering, POSTECH,

{kkm959595, choi.youngbin, seoyeon26, suhyeonjun, dongwoo.kim, sangdon}@postech.ac.kr

## Abstract

Tool-integrated LLM agents are often wrapped within a *harness*: the scaffolding that determines which tools are exposed, how they are described, and what auxiliary information accompanies each per-step observation. While agents are routinely post-trained, this scaffolding is typically treated as a *fixed* engineering detail, with design effort limited to the training-free regime. Moreover, existing post-training algorithms assume a static environment, even though tool environments and tasks often shift upon deployment. To address this gap, we extend ALFWorld (i) to treat the harness as a controllable design dimension and (ii) to support evaluation under task and tool environment shifts. Building on this, we systematically analyze how the harness design influences post-training in both in-distribution and out-of-distribution (OOD) settings. We empirically show that *harness-aware post-training* not only improves in-distribution performance but also enables agents to robustly adapt to OOD settings. Under a harness with minimal design effort, post-training suffers a drastic performance drop under stronger tool environment shifts, further highlighting the importance of harness-aware post-training under such shifts.

## 1 Introduction

LLM agents are designed to solve complex problems that often require multi-step reasoning (Yao et al., 2023; Shinn et al., 2023; Wang et al., 2024; Yang et al., 2024; Wu et al., 2025). Among them, *tool-integrated agentic systems* extend LLM capabilities by proactively interacting with environments through predefined tools (Schick et al., 2023; Qin et al., 2024; Patil et al., 2024; Liu et al., 2024). Accordingly, performance depends not only on producing correct final outputs but also on appropriately invoking tools, which has motivated increas-

ing efforts to post-train pretrained LLMs for effective tool use (Feng et al., 2026).

A factor that is often overlooked in this line of work is the *harness*: the code that wraps the LLM, determining what is stored, retrieved, and presented to it, and what is done with its outputs (Wang et al., 2025; Anthropic, 2026; OpenAI, 2026). While the harness is, in principle, an engineering detail, in practice it is a major determinant of overall performance: even with the same task and model, two harnesses that differ only in how they present auxiliary information can produce drastically different success rates (Yang et al., 2024; Badertdinov et al., 2025). For this reason, harness design has attracted increasing attention, ranging from manually curated prompts (Hong et al., 2024; Antoniadis et al., 2025; Lin et al., 2026) to recent attempts at automating the design itself (Lee et al., 2026; Shang et al., 2025). All of these efforts, however, are limited to a training-free regime, treating the harness as a way to elicit better performance, typically from a closed-source model. How harness design interacts with post-training, in contrast, remains largely unexamined.

In parallel, most existing post-training approaches for tool-integrated agents implicitly assume a static deployment scenario, in which both the task distribution and the tool environment, defined as the set of tools and their invocation protocols, remain fixed (Jin et al., 2025; Mai et al., 2025; Xue et al., 2026; Qian et al., 2025; Feng et al., 2025; Jiang et al., 2025; Cheng et al., 2025). In practice, this assumption is often violated: real-world deployments expose agents to shifts in both the task distribution and the tool environment. A first source of shift arises from changes in the task distribution, a standard out-of-distribution (OOD) problem at the input level (Koh et al., 2021; Yuan et al., 2023; Yang et al., 2023). For example, an agent originally designed for workload scheduling may later be repurposed for itinerary planning,

\*Equal contribution.

†Co-advising.

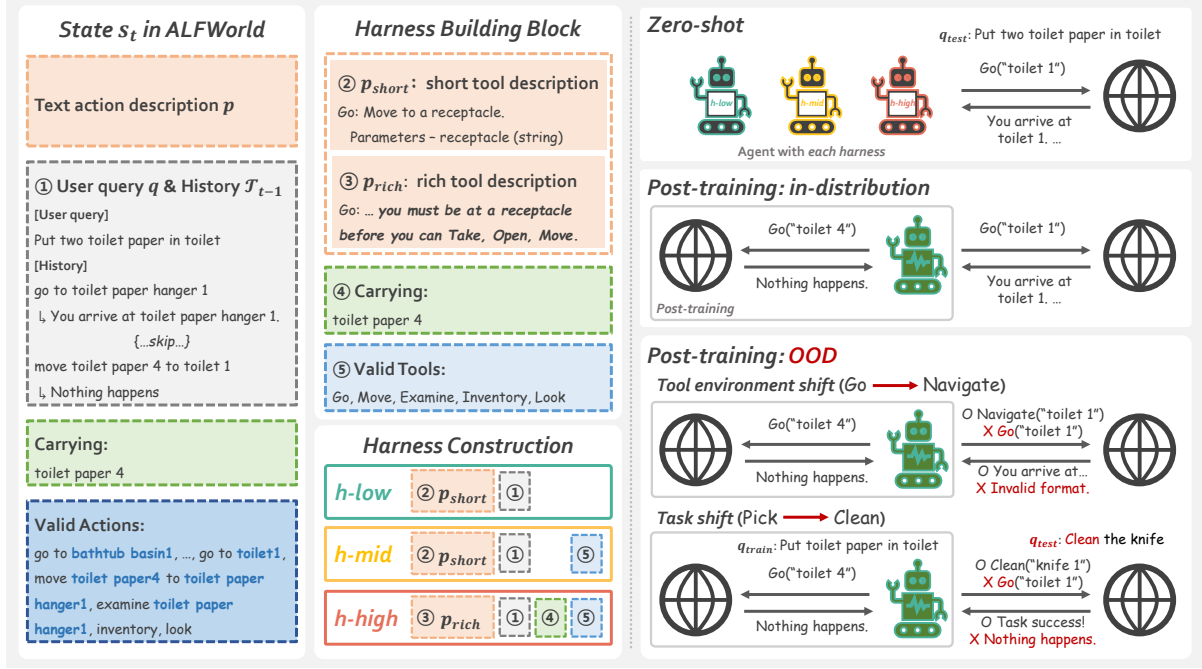


Figure 1: Overview of experimental design, built on our extension of ALFWorld (Shridhar et al., 2021) tailored to tool-integrated agentic tasks. (Left) State of the original text-based ALFWorld environment. (Middle) Harness building blocks for tool-integrated agentic tasks: different combinations of these blocks yield harnesses with varying levels of informativeness. (Right) Agents under each harness are evaluated in zero-shot and post-training regimes, where the latter spans in-distribution and OOD settings.

shifting objectives, constraints, and tool usage patterns, which we refer to as *task shift*. A second, more subtle source stems from changes in the tool environment itself: tools may be updated, modifying invocation protocols while preserving functionality, which we refer to as *tool environment shift*. This introduces a distinct OOD setting in which the task distribution remains fixed but the action interface itself changes, and it remains comparatively underexplored (Chen et al., 2025; Li et al., 2026).

In this paper, we ask the following question: *how does harness design influence post-training LLM agents in both in-distribution and OOD settings?* To this end, we extend ALFWorld (Shridhar et al., 2021), a text-based planning environment, into a benchmark tailored to tool-integrated agentic tasks. Specifically, we (i) treat the harness as a controllable design dimension and (ii) modularize the benchmark to support evaluation both in-distribution and under tool environment shift and task shift settings (Fig. 1). While existing post-training algorithms attain high success rates on ALFWorld (Feng et al., 2025), we observe that this performance largely stems from *harness-aware post-training*, where a specific harness is fixed a priori (Fig. 3). In particular, the original bench-

mark provides the agent with the complete set of feasible tool calls at every step, which constitutes a highly informative harness. Such a harness, however, comes at a cost: it can be constructed only when the transition dynamics of the ALFWorld environment are fully specified. Yet this cost is rarely made explicit in prior work. To address this gap, we provide three harnesses with varying levels of informativeness and empirically examine *whether the additional cost of harness design is indispensable for post-training*.

We empirically show that harness-aware post-training not only improves in-distribution performance but also enables agents to robustly adapt to OOD settings. Notably, under the stronger tool environment shift where only the invocation protocol changes, models post-trained on a harness with low design effort suffer from a drastic performance drop, despite their decent in-distribution performance, which is even comparable to that of the agents post-trained with an informative harness. In addition, we observe that applying an informative harness only after post-training under a low-effort harness yields performance inferior to that obtained when the harness is in place during post-training. Unlike the closed-source setting,

where post-hoc harness engineering is the standard practice (Lee et al., 2026), this result provides an actionable insight: a high-quality harness should be designed in advance and treated as an explicit part of the post-training recipe rather than as an inference-time add-on.

## 2 Preliminaries

We outline the formulation of multi-step tool-integrated agentic tasks and representative post-training methods, along with the notation used throughout the paper.

### 2.1 Multi-step Tool-integrated Agentic Tasks

Let  $\mathcal{V}$  be a token vocabulary,  $\mathcal{S} := \mathcal{V}^*$  be a state space, and  $\mathcal{A} := \mathcal{V}^*$  be an action space. Given a system prompt  $p \in \mathcal{V}^*$  and a user query  $q \in \mathcal{V}^*$ , an LLM agent  $\pi : \mathcal{S} \rightarrow \mathcal{A}$  generates a sequence of action tokens  $a_t := \pi(s_t) \in \mathcal{A}$  at each time step  $t \in \{1, \dots, T\}$ . Here,  $s_t \in \mathcal{S}$  is the state at step  $t$  provided to the agent, and  $T$  is the total number of tool calling steps. Specifically,  $s_t := (p, q, \mathcal{T}_{t-1})$ , where  $\mathcal{T}_{t-1}$  is a history of tool calling steps up to the  $(t-1)$ -th step, and  $\mathcal{T}_0$  is an empty sequence.

The tool environment  $\text{TE} : \mathcal{V}^* \rightarrow \mathcal{V}^*$  then plays two roles: in the *tool calling phase*, it returns the tool calling result  $e_t := \text{TE}((s_t, a_t))$ ; in the *state reconstruction phase*, it aggregates  $e_t$  with  $a_t$  and  $\mathcal{T}_{t-1}$  into a history  $\mathcal{T}_t := \text{TE}((\mathcal{T}_{t-1}, a_t, e_t))$ , provided to the agent as the next state  $s_{t+1} = (p, q, \mathcal{T}_t)$ . As such, the  $t$ -th tool calling step can be represented as:

$$\begin{array}{c} s_t = (p, q, \mathcal{T}_{t-1}) \xrightarrow{+a_t=\pi(s_t)} (p, q, \mathcal{T}_{t-1}, a_t) \xrightarrow{+e_t=\text{TE}((s_t, a_t))} \\ \underbrace{\hspace{10em}}_{\text{Tool Calling Phase}} \\ \rightarrow (p, q, \mathcal{T}_{t-1}, a_t, e_t) \xrightarrow[\substack{-(\mathcal{T}_{t-1}, a_t, e_t) \\ +\mathcal{T}_t=\text{TE}((\mathcal{T}_{t-1}, a_t, e_t))}]{\hspace{1em}} s_{t+1} = (p, q, \mathcal{T}_t). \\ \underbrace{\hspace{10em}}_{\text{State Reconstruction Phase}} \end{array}$$

**Remark.** While “harness” generally refers to any wrapper around the agent (Lin et al., 2026), we restrict it to wrappers acting through  $p$  and  $\text{TE}$ . These are the core components of the agentic workflow that shape what the agent observes in our environment. As detailed in §3, harness, tool schema, and task type jointly affect the transition dynamics of the agentic workflow. In particular, tool environment shift and task shift pose challenges for training an agent that remains robust to deployment-time workflows differing from those encountered during training.

### 2.2 Training Tool-integrated Agentic Systems

We model a multi-step tool calling process as a Markov decision process (MDP) at a sequence level. The goal is to find a policy  $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$ , parameterized by  $\theta \in \Theta$ , that maximizes  $\mathcal{J}(\theta) = \mathbb{E}[R(\tau)]$ , where  $\tau := \{(s_1, a_1, s_2), \dots, (s_T, a_T, s_{T+1})\}$  is an episode-level trajectory and  $R : (\mathcal{S} \times \mathcal{A} \times \mathcal{S})^T \rightarrow \mathbb{R}$  is an *episode-level* reward function. A common choice is  $R(\tau) := \sum_{t=1}^T \gamma^{t-1} r_t$ , where  $\gamma \in [0, 1]$  is a discount factor and  $r_t := r(s_t, a_t, s_{t+1})$  is a *step-level* reward.

Group-relative policy optimization (GRPO) (Shao et al., 2024) is a widely adopted algorithm for post-training LLMs due to its computational efficiency in estimating  $\nabla_\theta \mathcal{J}(\theta)$ . However, as GRPO uniformly assigns the episode-level advantage across all steps, it suffers from a credit assignment problem in long-horizon tasks (Dong et al., 2026; Fang et al., 2026). Unlike GRPO, group-in-group policy optimization (GiGPO) (Feng et al., 2025) differentiates the advantage assigned to each step within the same episode by introducing a step-level advantage on top of the episode-level advantage. Yet both algorithms assume a static agentic workflow. A formal description of both algorithms is provided in Appendix C.

**Remark.** Let  $\mathcal{H}$  be a harness space. Given a harness  $h \in \mathcal{H}$  specified by  $(p, \text{TE})$ , an agent  $\pi_\theta$  is post-trained using trajectories whose per-step states and transitions are induced by the same harness  $h$ . Concretely, the agent is post-trained on states constructed by that harness. *Harness-aware post-training* is then defined as post-training an LLM agent using an “informative” harness that encodes rich prior knowledge about the environment or the task itself, often requiring costly design effort in advance. Existing works that adopt the aforementioned algorithms in ALFWorld fix a highly informative harness in advance, rather than treating it as a design variable. As such, they lack a controlled analysis of how harness-aware post-training affects performance and OOD robustness of LLM agents.

## 3 Experiment Setup

### 3.1 ALFWorld for Tool-based Agentic Tasks

ALFWorld (Shridhar et al., 2021) is a text-based planning environment with 3,827 task instances across six categories of common household activities: Pick & Place (Pick), Examine in Light (Look),

| Harness Ver. | Tool Description<br>in $p$ | Valid tools:<br>(in $\mathcal{T}_t$ ) | Carrying:<br>(in $\mathcal{T}_t$ ) |
|--------------|----------------------------|---------------------------------------|------------------------------------|
| h-low        | Short                      | –                                     | –                                  |
| h-mid        | Short                      | ✓                                     | –                                  |
| h-high       | Rich                       | ✓                                     | ✓                                  |

Table 1: Each harness version, characterized by how it shapes the system prompt  $p$  and the per-step history  $\mathcal{T}_t$  produced by TE. Valid tools: and Carrying: are auxiliary lines appended to  $\mathcal{T}_t$ , listing the tools admissible at the current state and the object the agent is currently carrying, respectively.

Clean & Place (Clean), Heat & Place (Heat), Cool & Place (Cool), and Pick Two & Place (Pick 2). Every task requires multi-step decision making, since it consists of a sequence of sub-goals. This structure makes ALFWorld a natural testbed for evaluating the planning capability of LLM agents.

We extend ALFWorld by reformulating its original tasks as tool-integrated reasoning problems. Specifically, text actions in ALFWorld can naturally be interpreted as tool calls: verbs correspond to tools and entities serve as arguments. For instance, a text action “Go to drawer 1” in the original ALFWorld environment is mapped to a tool call “Go(receptacle=‘drawer 1’)”.

In subsequent sections, we describe how our extended benchmark enables a controlled study of harness design in post-training LLM agents by modularizing three components: harness (§3.2), tool schema (§3.3), and task type (§3.4).

### 3.2 Harness

In tool-integrated agentic tasks, the harness determines how much guidance the agent receives about using the available tools. To this end, we instantiate three harnesses with varying degrees of informativeness, each building on the previous one: h-low, h-mid, and h-high.

The base version, h-low, is designed with low design effort: every tool carries only a one-line description in  $p$  (e.g., “Move to a receptacle” for Go), and each per-step history  $\mathcal{T}_t$  contains nothing beyond the raw observation. h-mid additionally augments every per-step history with the set of tools admissible at the current state, leaving the one-line descriptions in  $p$  unchanged. Finally, h-high further expands each tool description in  $p$  into a richer form, covering its preconditions, interactions with other tools, and role in completing tasks. It also appends the object the agent is cur-

| Tool Schema | Example                                                       |
|-------------|---------------------------------------------------------------|
| v1.0        | Go(receptacle=“drawer 1”)                                     |
| v1.1        | NavigateTo(destination=“drawer 1”)                            |
| v2.0        | ReceptacleControl(action=“navigate_to”,<br>target=“drawer 1”) |

Table 2: Tool schema versions and the valid form of tool calls for the same operation “move to drawer 1” under each schema. v1.1 applies paraphrasing to v1.0, while v2.0 additionally groups tools by structural and functional similarity.

rently carrying to each per-step history, which it would otherwise have to retrieve through a separate Inventory call. Table 1 summarizes the three versions, and the corresponding tool descriptions and per-step histories are deferred to Appendix B.1.

Informative harnesses like h-mid and h-high, however, do not come for free. They must either be hand-designed by experts with sufficient prior knowledge of the environment, or be discovered through costly exploratory interaction with it. This cost is rarely made explicit. Most prior studies on ALFWorld in fact operate under a specific harness that goes beyond even h-high, providing the agent with the full set of feasible tool calls at every step, yet this is rarely flagged as an assumption. We instead treat the harness as a controllable variable, with h-low, h-mid, and h-high corresponding to three distinct levels of informativeness. We acknowledge that h-mid and h-high expose environment-derived information that would otherwise require costly exploration to obtain. Indeed, the “Valid tools:” field is an oracle-style admissibility signal computed from the ALFWorld simulator. The “Carrying:” field is likewise state-dependent, though it could in principle be maintained by tracking previous interactions. In addition, the rich tool descriptions in  $p$  provide state-independent guidelines, which would nonetheless require expert knowledge or a sufficient amount of exploration to obtain. *We view h-mid and h-high as representing harnesses obtained through different amounts of such exploration, treating informativeness itself as a design choice that shapes agent performance (RQ1 in §4).*

### 3.3 Tool Schema

The tool schema determines the surface form of admissible tool calls. Taking v1.0 as the base schema, we provide two environment shift scenarios of varying degrees, where the stronger shift (v2.0) ex-

tends the milder one (v1.1) through additional updates to the tool invocation protocols. Specifically, v1.0 is the base schema itself, consisting of 13 tools defined in ALFWorld with intuitive verb-style names such as Go, Take, Heat, Slice, and Look. The second, v1.1, applies paraphrasing: a semantics-preserving renaming of every tool name and parameter key from v1.0 into more descriptive forms (e.g., Go  $\rightarrow$  NavigateTo, with its parameter receptacle  $\rightarrow$  destination), leaving the cardinality of the tool set unchanged. The third, v2.0, additionally applies grouping by structural and functional similarity: building on v1.1, tools that share structural or functional roles are consolidated into a smaller, higher-level set, reducing the cardinality from 13 to 5. Each consolidated tool exposes its sub-operations through a discrete action parameter. For example, NavigateTo, OpenContainer, and CloseContainer are merged into a single ReceptacleControl tool, whose action parameter takes one of three values: navigate\_to, open\_container, or close\_container. Table 2 provides an illustrative example of the three versions, with full details in Appendix B.2.

Across the three versions, both  $p$  and TE are updated accordingly, with details deferred to Appendix B.2. Tool names, parameter keys, and descriptions in  $p$  are revised to match the active schema, while the overall structure of  $p$  is preserved. TE’s validation logic is adjusted accordingly. As an illustration, the same operation “move to drawer 1” issued at step  $t$  takes a distinct valid form under each schema (Table 2). A tool call has a valid format only under the schema in which it is defined. Otherwise, even though the call expresses the same operation, TE returns  $e_t = \text{“Invalid tool format”}$ . We aim to analyze *whether harness-aware post-training enables agents to robustly adapt to such shifts in tool invocation protocols* (RQ3 in §4).

### 3.4 Task Type

The task type determines the distribution from which the user query  $q$  is drawn. We group the six task categories of ALFWorld by the minimum number of sub-goals required for completion into three difficulty levels: t-easy, t-med, and t-hard (Table 3). For example, task instances in the Pick category (e.g., “Put a plate on the coffee table”) are categorized as t-easy, requiring a sequence of four sub-goals: find an object (“plate”), pick it up, find the correct location (“coffee table”)

| Group  | Task   | # Sub-goals | Example                                  |
|--------|--------|-------------|------------------------------------------|
| t-easy | Pick   | 4           | “Put a plate on the coffee table”        |
|        | Look   | 3           | “Examine a book under the lamp”          |
| t-med  | Clean  | 5           | “Clean the knife and put in the drawer”  |
|        | Heat   | 5           | “Heat a mug and put on the coffee table” |
|        | Cool   | 5           | “Put a cool bottle on the countertop”    |
| t-hard | Pick 2 | 8           | “Put two pencils in the drawer”          |

Table 3: Six task categories of ALFWorld grouped into three difficulty levels (t-easy, t-med, t-hard) by the minimum number of sub-goals required for completion.

to place it, and put it down there. This grouping serves as the basis for the task shift regime: training on one difficulty level and evaluating on the others constitutes a controlled shift in the input distribution, with the tool schema held fixed.

While task instances are categorized into three task groups, some sub-goals are shared across instances from different groups. For example, every task instance shares the same initial sub-goal: to find an object of the desired type. Because the corresponding tool calls are reused across instances (possibly with different arguments), an ideal post-training method should learn to generate such tool calls in a way that generalizes to task instances unseen during training, rather than overfitting to the training distribution. While these household activities are simple from a human perspective, results in §4 show that they present non-trivial challenges to LLM agents in both in-distribution and OOD scenarios. We further ask *whether the prior knowledge encoded in an informative harness helps an agent generalize across task groups* (RQ3 in §4).

## 4 Analysis

In this section, we present a controlled analysis of how harness design influences the post-training of LLM agents, both in-distribution and under OOD settings. Specifically, we investigate the following research questions:

- **RQ1:** Does the influence of harness informativeness, observed at zero-shot (Obs. 1), extend to harness-aware post-training (Obs. 2)?
- **RQ2:** Can a harness be applied only after post-training, or should it be in place during training (Obs. 3)?
- **RQ3:** Does harness-aware post-training enable robustness to tool environment shift (Obs. 4) and task shift (Obs. 5)?

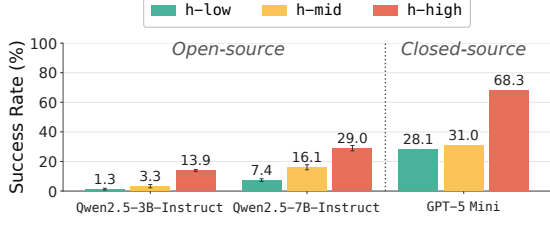


Figure 2: Zero-shot success rates without post-training: Pretrained agents are evaluated on  $\mathcal{D}_{\text{all}}^{\text{te}}$  under each harness with tool schema fixed to v1.0. Zero-shot performance improves with harness informativeness, more pronounced for high-capacity models.

#### 4.1 Evaluation Protocol

Among 3,827 task instances in ALFWorld, 3,553 form the training set  $\mathcal{D}_{\text{all}}^{\text{tr}}$  and the remaining 274 form the test set  $\mathcal{D}_{\text{all}}^{\text{te}}$ . The test set corresponds to the standard seen/unseen split of ALFWorld (140 seen and 134 unseen instances). Both sets are partitioned by task difficulty:  $\mathcal{D}_{\text{all}}^{\text{tr}} := \mathcal{D}_{\text{easy}}^{\text{tr}} \cup \mathcal{D}_{\text{med}}^{\text{tr}} \cup \mathcal{D}_{\text{hard}}^{\text{tr}}$  and  $\mathcal{D}_{\text{all}}^{\text{te}} := \mathcal{D}_{\text{easy}}^{\text{te}} \cup \mathcal{D}_{\text{med}}^{\text{te}} \cup \mathcal{D}_{\text{hard}}^{\text{te}}$ . Here,  $\mathcal{D}_{\text{easy}}^{\text{tr}}$ ,  $\mathcal{D}_{\text{med}}^{\text{tr}}$ ,  $\mathcal{D}_{\text{hard}}^{\text{tr}}$  refer to the training instances within each difficulty-based task group (t-easy, t-med, t-hard) defined in §3.4. The test splits are defined analogously. We report the *success rate*, defined as the fraction of test task instances in which the agent completes all required sub-goals within the maximum number of tool calling steps.

For *zero-shot analysis*, we evaluate pretrained LLM agents directly under each of the three harnesses {h-low, h-mid, h-high} with tool schema fixed to v1.0 (Obs. 1). Specifically, we evaluate GPT-5 Mini as a closed-source model, and instruction-tuned versions of Qwen2.5-3B and Qwen2.5-7B as open-source models. For GPT-5 Mini, we set its reasoning effort to “high” to use the most capable inference setting.

For *post-training analysis*, we post-train the same open-source models with GRPO and GiGPO as RL algorithms. For in-distribution and tool environment shift scenarios, agents are post-trained on the full training split  $\mathcal{D}_{\text{all}}^{\text{tr}}$  under each of the three harnesses, with tool schema fixed to v1.0. Post-trained agents are then evaluated on  $\mathcal{D}_{\text{all}}^{\text{te}}$  under the same harness, either with schema v1.0 (in-distribution, Obs. 2) or with v1.1 and v2.0 corresponding to tool environment shifts of varying degrees (Obs. 4). For the task shift scenario, agents are instead post-trained on a single difficulty-specific split ( $\mathcal{D}_{\text{easy}}^{\text{tr}}$ ,  $\mathcal{D}_{\text{med}}^{\text{tr}}$ , or  $\mathcal{D}_{\text{hard}}^{\text{tr}}$ ) under each of the three harnesses, with tool schema fixed to v1.0,

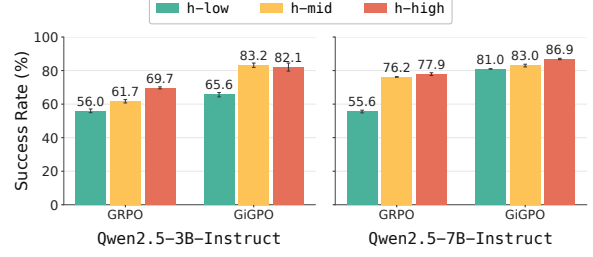


Figure 3: In-distribution success rates after post-training: Agents are post-trained on  $\mathcal{D}_{\text{all}}^{\text{tr}}$  under each harness with tool schema fixed to v1.0, and evaluated on  $\mathcal{D}_{\text{all}}^{\text{te}}$  under the same harness and schema. Post-training performance tends to improve with harness informativeness.

and evaluated on the test splits of the remaining groups under the same harness and schema (Obs. 5). All open-source results are averaged over three random seeds, with standard deviations reported in the corresponding figures and tables. Additional training and evaluation details are provided in Appendix C and Appendix D.

**Remark.** Prior post-training studies on ALFWorld typically operate under a single (harness, task type) configuration, training only on the full training split  $\mathcal{D}_{\text{all}}^{\text{tr}}$ . We instead treat harness and task type as controllable variables, running each RL algorithm across all combinations of three harness levels and four task type settings (the full training split and three difficulty-specific splits), for both open-source models. This main grid of experiments alone amounts to approximately 1,800 H200 GPU-hours.

#### 4.2 Zero-shot and In-distribution Performance (RQ1)

**Observation 1 (Zero-shot).** *Harness informativeness monotonically improves zero-shot performance of LLM agents, and the magnitude of the gain scales with model capacity.*

Across all three models, performance improves monotonically with harness informativeness (Fig. 2). In particular, GPT-5 Mini shows the largest gain, confirming that the magnitude of the gain scales with model capacity. Furthermore, while open-source models fail at Pick 2 (t-hard), reporting 0.0 across most harness configurations, GPT-5 Mini achieves 61.0 under h-high (Table 10). Even under h-low, GPT-5 Mini achieves 17.1 at Pick 2, further illustrating that model capacity is essential to drive harness-induced gains.

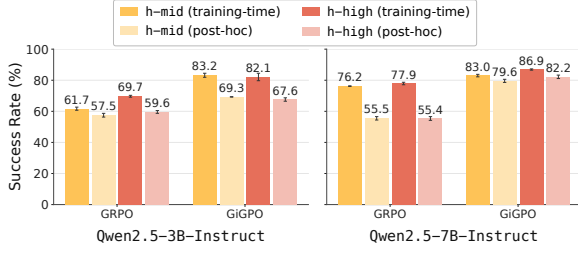


Figure 4: Training-time vs. post-hoc harness application: in-distribution success rates on  $\mathcal{D}_{\text{all}}^{\text{le}}$ . For each harness  $\{h\text{-mid}, h\text{-high}\}$ , we compare an agent post-trained with that harness in place (*training-time*) against an agent post-trained under  $h\text{-low}$  with that harness applied only at evaluation (*post-hoc*). Training-time application consistently outperforms post-hoc application.

**Observation 2 (In-distribution).** *The monotonic harness gain observed at zero-shot (Fig. 2) largely carries over after post-training under both algorithms.*

The monotonic harness gain observed at zero-shot extends to post-training under both algorithms (Fig. 3). For example, Qwen2.5-3B-Instruct post-trained with GRPO under  $h\text{-high}$  outperforms Qwen2.5-7B-Instruct post-trained with GRPO under  $h\text{-low}$  by 14.1 points, indicating that the choice of harness can outweigh the effect of model capacity even after post-training. We also observe that GiGPO consistently outperforms GRPO across all configurations, consistent with its finer credit assignment in long-horizon tasks.

### 4.3 Post-hoc Harness Application (RQ2)

**Observation 3 (Post-hoc Gap).** *Applying a harness only after training recovers little of the benefit of training with it in place.*

Once the harness is treated as a controllable variable, a natural question arises: should an informative harness be applied at training time, or in a post-hoc manner after post-training under a low-effort harness? For closed-source models, post-hoc harness engineering is the default practice (Lee et al., 2026), so one might expect the same recipe to suffice for open-source models after post-training as well. Our results show that it does not: we find that harness-aware post-training is preferable across all model and harness configurations (Fig. 4). The gap is particularly large for Qwen2.5-7B-Instruct post-trained with GRPO, where training-time harness application outperforms post-hoc application by 20.7 points under  $h\text{-mid}$  and 22.5 points under  $h\text{-high}$ .

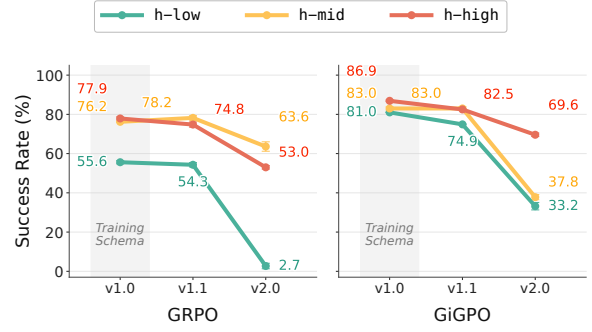


Figure 5: Post-training success rates of Qwen2.5-7B-Instruct under tool environment shift: Agents are post-trained on  $\mathcal{D}_{\text{all}}^{\text{tr}}$  under each harness with tool schema fixed to  $v1.0$ , and evaluated on  $\mathcal{D}_{\text{all}}^{\text{le}}$  under the same harness but with schema  $v1.1$  (mild shift) or  $v2.0$  (stronger shift). Across both algorithms, more informative harnesses generally yield greater robustness to such shifts.

In addition, we conduct a short adaptation experiment: Starting from the model checkpoint post-trained with GiGPO under  $h\text{-low}$  (Qwen2.5-3B-Instruct), we continue post-training under  $h\text{-mid}$  or  $h\text{-high}$  using 20% of the main training budget. Short adaptation improves over naive post-hoc application, from 69.3 to 71.5 under  $h\text{-mid}$  and from 67.6 to 75.9 under  $h\text{-high}$ , but agents post-trained with the target harness from the outset remain the strongest, leaving gaps of 11.7 and 6.2 points. The same trend holds for GRPO (Table 18). Short adaptation thus partially closes the post-hoc gap, while early alignment between the harness and post-training remains preferable. This suggests that the harness is better specified before post-training, allowing the agent to adapt to the harness it will ultimately use.

### 4.4 OOD Robustness (RQ3)

**Observation 4 (Tool Environment Shift).** *Harness-aware post-training mitigates the performance drop under tool environment shift, while post-training under  $h\text{-low}$  with low design effort suffers a drastic drop under stronger shift.*

Harness-aware post-training not only tends to improve in-distribution performance (Obs. 2), but also mitigates the performance drop under tool environment shift (Fig. 5). While the agent post-trained under a harness with low design effort ( $h\text{-low}$ ) adapts to the mild shift ( $v1.1$ ), it suffers a drastic performance drop under  $v2.0$ . In particular, Qwen2.5-7B-Instruct post-trained with GRPO under  $h\text{-low}$  achieves 2.7 under  $v2.0$ , which is 10.8 points below the base model without post-

| Harness | Training-time Form (v1.0)      | Test-time Form (v2.0)                                           | Agent Output                     |
|---------|--------------------------------|-----------------------------------------------------------------|----------------------------------|
| h-low   | Go(receptacle="countertop 1")  | ReceptacleControl(action="navigate_to", target="countertop 1")  | GoTo("countertop 1")             |
|         | Go(receptacle="shelf 1")       | ReceptacleControl(action="navigate_to", target="shelf 1")       | GoToLocation("shelf 1")          |
|         | Go(receptacle="diningtable 1") | ReceptacleControl(action="navigate_to", target="diningtable 1") | GoToDiningTable("diningtable 1") |
| h-mid   | Go(receptacle="countertop 1")  | ReceptacleControl(action="navigate_to", target="countertop 1")  | navigate_to("countertop 1")      |
|         | Go(receptacle="dresser 1")     | ReceptacleControl(action="navigate_to", target="dresser 1")     | go_to("dresser 1")               |
|         | Go(receptacle="dresser 1")     | ReceptacleControl(action="navigate_to", target="dresser 1")     | goto_dresser()                   |

Table 4: Example failure modes of Qwen2.5-7B-Instruct post-trained with GiGPO under h-low and h-mid, with tool schema v1.0 at training time and v2.0 at evaluation. The corresponding valid forms under v1.0 and v2.0 are shown for reference. The agents fail to adapt to the test-time schema, generating plausible-looking but non-existent tool names.

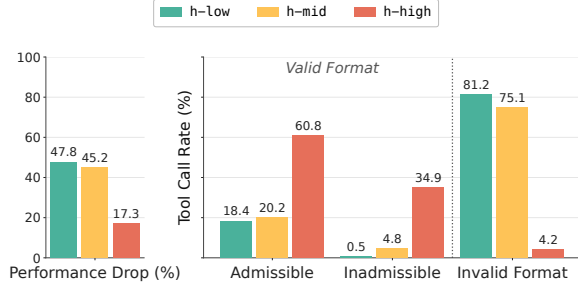


Figure 6: In-depth analysis of post-training under the tool shift from v1.0 (training) to v2.0 (evaluation). (Left) Performance drop under v2.0 relative to that of in-distribution. (Right) Breakdown of generated tool calls into three categories: *admissible* (correct format and executable), *inadmissible* (correct format but rejected by the environment, i.e., “Nothing happens”), and *invalid* (malformed, e.g., non-existent tool name or argument errors).

training (13.5, Table 13).

While Qwen2.5-7B-Instruct post-trained with GiGPO under h-low performs well in-distribution (Fig. 3), it exhibits a substantial drop under v2.0 (Fig. 5), indicating that the agent fails to adapt to the test-time schema. As such, we conduct an in-depth analysis on the tool calling patterns of Qwen2.5-7B-Instruct post-trained with GiGPO, where tool schema is fixed to v1.0 at training time and v2.0 at evaluation.

The model post-trained under h-high generates 95.7% valid tool calls under v2.0, indicating that it adapts well to the tool environment shift at the surface level (Fig. 6). However, 34.9% of these calls are not admissible at the current state, resulting in a performance drop of 17.3 points (from 86.9 to 69.6) relative to its in-distribution result. By contrast, the models post-trained under h-mid and h-low fail to adapt at the schema level, returning  $e_t$  = “Invalid tool format” in 75.1% and 81.2% of attempts, respectively. Performance degrades by 45.2 and 47.8 points under v2.0, sig-

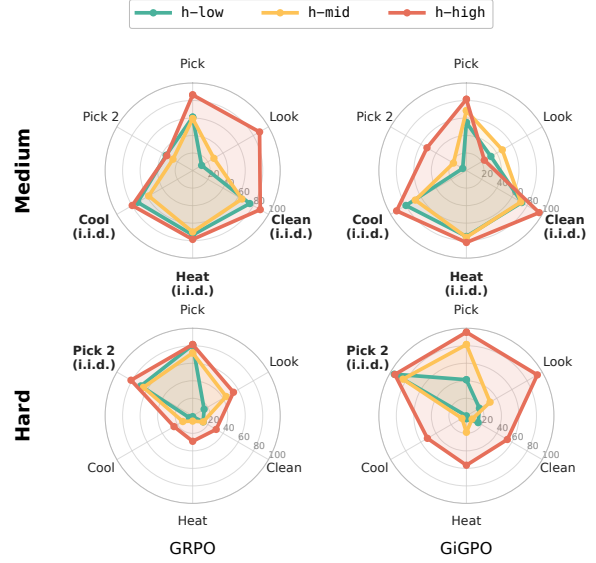


Figure 7: Post-training success rates of Qwen2.5-7B-Instruct under task shift: agents are post-trained on  $\mathcal{D}_{\text{med}}^{\text{tr}}$  (top row) and  $\mathcal{D}_{\text{hard}}^{\text{tr}}$  (bottom row) under each harness with tool schema fixed to v1.0, and evaluated on  $\mathcal{D}_{\text{all}}^{\text{tr}}$  under the same harness and schema. In-distribution task categories are highlighted in **bold**: **Clean**, **Heat**, **Cool** for  $\mathcal{D}_{\text{med}}^{\text{tr}}$ , and **Pick 2** for  $\mathcal{D}_{\text{hard}}^{\text{tr}}$ . Post-training under informative harnesses tends to improve performance on OOD task categories.

nificantly larger than that of h-high. This underscores the importance of training with an informative harness for OOD robustness, particularly under stronger tool environment shift. Representative failure modes are provided in Table 4.

**Observation 5 (Task Shift).** *Prior knowledge encoded in the harness boosts inter-task transfer of agent performance.*

Performance generally improves with harness informativeness in the task shift scenario as well, where this trend holds across both algorithms (Fig. 7). GRPO improves from 33.1 under h-low to 50.6 under h-high (+17.5), and GiGPO from 28.8 to 73.4 (+44.6, Table 16). This improvement

arises not only from the in-distribution task category (Pick 2), but also from the OOD categories, where agents post-trained under h-low fail to transfer (Fig. 7).

Specifically, Qwen2.5-7B-Instruct post-trained with GRPO under h-low achieves 0.9 on Heat, which rises to 29.1 under h-high (Table 16). A similar pattern holds with GiGPO under h-low, which achieves 0.0 on Cool and rises to 51.4 under h-high. This indicates that the prior knowledge encoded in an informative harness helps the agent generalize across task groups. Similar trends are observed in additional experiments under fine-grained task shift settings, in which agents are post-trained on a single task category (Table 19). As in the tool environment shift scenario, harness-aware post-training improves OOD robustness.

## 5 Conclusion

In this paper, we studied harness design in the context of post-training tool-integrated LLM agents, a factor that prior work has either treated as a fixed engineering detail or examined only in the training-free regime. Building on ALFWorld, we constructed a benchmark that treats the harness as a controllable design dimension and supports four evaluation regimes: zero-shot, and post-training under the in-distribution and two OOD scenarios (tool environment shift and task shift). Across these regimes, we instantiated three harnesses with varying levels of informativeness and analyzed how harness design influences post-training of LLM agents in both in-distribution and OOD settings.

Our experiments establish that harness design and post-training cannot be treated as separable design choices. Performance improves monotonically with harness informativeness at zero-shot, and this trend largely carries over to post-training under the in-distribution scenario. This improvement, however, requires the harness to be in place during post-training: applying it only afterward recovers little of the benefit of training with one in place. Under OOD settings, harness-aware post-training mitigates the performance drop under tool environment shift and helps the agent generalize across task groups, whereas post-training under a harness with low design effort suffers a drastic performance drop under stronger tool environment shift and fails to transfer to OOD task categories. These results demonstrate that harness-aware post-training is a

prerequisite for, rather than a supplement to, robust performance of tool-integrated LLM agents.

## Limitations

Our main analysis is conducted on a single environment, ALFWorld. Although our benchmark built on ALFWorld enables a controlled study of harness design under in-distribution and OOD settings, validating our findings on benchmarks with different task distributions and tool environments is an important direction. To partially address concerns about the generalizability of our findings, we replicate our tool environment shift protocol on WebShop (Yao et al., 2022), an interactive e-commerce benchmark. The central pattern reproduces: the agent post-trained under the more informative harness consistently outperforms its low-effort counterpart under the shifted tool schema with both GRPO and GiGPO, and the agent post-trained under the low-effort harness collapses under GiGPO (success rate  $54.7 \rightarrow 33.5$ ) despite reasonable in-distribution performance (Table 17).

We examine harness-aware post-training with two open-source models, Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct, and two RL algorithms, GRPO and GiGPO, spanning 24 training configurations per model and amounting to approximately 1,800 H200 GPU-hours for the main ALFWorld experiments. Extending this to a broader range of models and RL algorithms, beyond what our compute budget allowed, is left to future work.

We instantiate three harnesses (h-low, h-mid, h-high) as representative levels of informativeness. In practice, informative harnesses such as h-mid and h-high must be obtained either through expert design or costly exploration of the environment (§3.2). While GiGPO under h-low performs comparably to harness-aware counterparts under the in-distribution scenario, it still breaks down under stronger tool environment shift, indicating that finer credit assignment alone cannot replace an informative harness. A natural direction is therefore to develop harness-aware post-training methods that jointly optimize the harness and the LLM agent. Our benchmark, which supports arbitrary harness designs, can serve as a testbed for such algorithms.

As a final remark, the scope of our findings is confined to small-to-medium-scale open-source LLMs post-trained for bounded, domain-specific tool-use workflows. We emphasize that this regime is of growing practical importance, particularly in

terms of cost efficiency: agentic workloads issue large numbers of repetitive tool calls, so a compact model post-trained for a specific workflow is substantially more economical to serve than a frontier model invoked at every step, while also meeting latency, privacy, and on-device constraints. This trend is illustrated by FunctionGemma, a 270M-parameter open model designed for customized local function calling (Black and Kumar, 2025), and by xLAM, a family of action models from Salesforce whose compact variants are post-trained specifically for function calling in agentic systems (Zhang et al., 2025). In such deployments, the harness and the post-trained model are co-designed for a fixed target workflow, which is the setting our study directly informs.

## Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2024-00337955); by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (RS-2019-II191906, Artificial Intelligence Graduate School Program (POSTECH); RS-2024-00457882, National AI Research Lab Project and Artificial Intelligence Research Hub Project; RS-2024-00509258 and RS-2024-00469482, Global AI Frontier Lab; RS-2025-00560062); and by IITP under the Leading Generative AI Human Resources Development (IITP-2026-RS-2026-25546560) grant funded by the Korea government (MSIT).

## References

- Anthropic. 2026. Effective Harnesses for Long-running Agents. <https://www.anthropic.com/engineering/effective-harnesses-for-long-running-agents>. Anthropic Engineering Blog.
- Antonis Antoniadis, Albert Örwall, Kexun Zhang, Yuxi Xie, Anirudh Goyal, and William Wang. 2025. SWE-Search: Enhancing Software Agents with Monte Carlo Tree Search and Iterative Refinement. In *ICLR*.
- Ibragim Badertdinov, Alexander Golubev, Maksim Nekrashevich, Anton Shevtsov, Simon Karasik, Andrei Andriushchenko, Maria Trofimova, Daria Litvinseva, and Boris Yangel. 2025. *SWE-Rebench: An Automated Pipeline for Task Collection and Decontaminated Evaluation of Software Engineering Agents*. In *NeurIPS*.
- Kat Black and Ravin Kumar. 2025. *FunctionGemma: Bringing Bespoke Function Calling to the Edge*.
- Guoxin Chen, Zhong Zhang, Xin Cong, Fangda Guo, Yesai Wu, Yankai Lin, Wenzheng Feng, and Yasheng Wang. 2025. *Learning Evolving Tools for Large Language Models*. In *ICLR*.
- Mingyue Cheng, Jie Ouyang, Shuo Yu, Ruirao Yan, Yucong Luo, Zirui Liu, Daoyu Wang, Qi Liu, and Enhong Chen. 2025. Agent-R1: Training Powerful LLM Agents with End-to-end Reinforcement Learning. *arXiv preprint arXiv:2511.14460*.
- Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. *SFT memorizes, RL generalizes: A Comparative Study of Foundation Model Post-training*. In *ICML*.
- Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2026. *Agentic Reinforced Policy Optimization*. In *ICLR*.
- Yangyi Fang, Jiaye Lin, Xiaoliang Fu, Cong Qin, Haolin Shi, Chang Liu, and Peilin Zhao. 2026. Proximity-based Multi-turn Optimization: Practical Credit Assignment for LLM Agent Training. In *ACL*.
- Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjuan Zhong. 2026. ReTool: Reinforcement Learning for Strategic Tool Use in LLMs. In *ICLR*.
- Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. *Group-in-group Policy Optimization for LLM Agent Training*. In *NeurIPS*.
- Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiaowu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024. *MetaGPT: Meta Programming for a Multi-agent Collaborative Framework*. In *ICLR*.
- Pengcheng Jiang, Xueqiang Xu, Jiacheng Lin, Jinfeng Xiao, Zifeng Wang, Jimeng Sun, and Jiawei Han. 2025. *s3: You Don't Need that Much Data to Train a Search Agent via RL*. In *EMNLP*.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Serkan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. *Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning*. In *COLM*.
- Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanus Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, and 4 others. 2021. WILDS: A Benchmark of In-the-wild Distribution Shifts. In *ICML*.

- Yoonho Lee, Roshen Nair, Qizheng Zhang, Kangwook Lee, Omar Khattab, and Chelsea Finn. 2026. [Meta-harness: End-to-end Optimization of Model Harnesses](#). In *COLM*.
- Guangrui Li, Yaochen Xie, Yi Liu, Ziwei Dong, Xingyuan Pan, Tianqi Zheng, Jason Choi, Michael J. Morais, Binit Jha, Shaunak Mishra, Bingrou Zhou, Chen Luo, Monica Xiao Cheng, and Dawn Song. 2026. [The World Won't Stay Still: Programmable Evolution for Agent Benchmarks](#). *arXiv preprint arXiv:2603.05910*.
- Jiahang Lin, Shichun Liu, Chengjun Pan, Lizhi Lin, Shihan Dou, Zhiheng Xi, Xuanjing Huang, Hang Yan, Zhenhua Han, Tao Gui, and 1 others. 2026. [Agentic Harness Engineering: Observability-driven Automatic Evolution of Coding-agent Harnesses](#). *arXiv preprint arXiv:2604.25850*.
- Zuxin Liu, Thai Hoang, Jianguo Zhang, Ming Zhu, Tian Lan, Shirley Kokane, Juntao Tan, Weiran Yao, Zhiwei Liu, Yihao Feng, Rithesh Murthy, Liangwei Yang, Silvio Savarese, Juan Carlos Niebles, Huan Wang, Shelby Heinecke, and Caiming Xiong. 2024. [APIGen: Automated Pipeline for Generating Verifiable and Diverse Function-calling Datasets](#). In *NeurIPS*.
- Xinji Mai, Haotian Xu, Xing W, Weinong Wang, Yingying Zhang, and Wenqiang Zhang. 2025. [Agentic RL Scaling law: Spontaneous Code Execution for Mathematical Problem Solving](#). In *NeurIPS*.
- Zhanfeng Mo, Xingxuan Li, Yuntao Chen, and Lidong Bing. 2025. [Multi-agent Tool-integrated Policy Optimization](#). *arXiv preprint arXiv:2510.04678*.
- OpenAI. 2026. [Harness Engineering: Leveraging Codex in an Agent-first World](#). <https://openai.com/index/harness-engineering/>. OpenAI Blog.
- Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E. Gonzalez. 2024. [Gorilla: Large Language Model Connected with Massive APIs](#). In *NeurIPS*.
- Cheng Qian, Emre Can Acikgoz, Qi He, Hongru WANG, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. 2025. [ToolRL: Reward is All Tool Learning Needs](#). In *NeurIPS*.
- Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. 2024. [ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs](#). In *ICLR*.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. [Toolformer: Language Models Can Teach Themselves to Use Tools](#). In *NeurIPS*.
- Yu Shang, Yu Li, Keyu Zhao, Likai Ma, Jiahe Liu, Fengli Xu, and Yong Li. 2025. [AgentSquare: Automatic LLM Agent Search in Modular Design Space](#). In *ICLR*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models](#). *arXiv preprint arXiv:2402.03300*.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. [Reflexion: Language Agents with Verbal Reinforcement Learning](#). In *NeurIPS*.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. [ALFWorld: Aligning Text and Embodied Environments for Interactive Learning](#). In *ICLR*.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. [Voyager: An Open-ended Embodied Agent with Large Language Models](#). *TMLR*.
- Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, and 5 others. 2025. [OpenHands: An Open Platform for AI Software Developers as Generalist Agents](#). In *ICLR*.
- Junde Wu, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. 2025. [Agentic Reasoning: A Streamlined Framework for Enhancing LLM Reasoning with Agentic Tools](#). In *ACL*.
- Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. 2026. [SimpleTIR: End-to-end Reinforcement Learning for Multi-turn Tool-integrated Reasoning](#). In *ICLR*.
- John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. 2024. [SWE-Agent: Agent-computer Interfaces Enable Automated Software Engineering](#). In *NeurIPS*.
- Linyi Yang, Yaoxian Song, Xuan Ren, Chenyang Lyu, Yidong Wang, Jingming Zhuo, Lingqiao Liu, Jindong Wang, Jennifer Foster, and Yue Zhang. 2023. [Out-of-distribution Generalization in Natural Language Processing: Past, Present, and Future](#). In *EMNLP*.
- Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. [WebShop: Towards Scalable Real-world Web Interaction with Grounded Language Agents](#). In *NeurIPS*.

- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. [ReACT: Synergizing Reasoning and Acting in Language Models](#). In *ICLR*.
- Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, FangYuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. 2023. [Revisiting out-of-distribution Robustness in NLP: Benchmarks, Analysis, and LLMs Evaluations](#). In *NeurIPS*.
- Jianguo Zhang, Tian Lan, Ming Zhu, Zuxin Liu, Thai Hoang, Shirley Kokane, Weiran Yao, Juntao Tan, Zhiwei Liu, Yihao Feng, and 1 others. 2025. xLAM: A Family of Large Action Models to Empower AI Agent Systems. In *NAACL*.
- Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, Jian Xie, Yuxuan Sun, Boyu Gou, Qi Qi, Zihang Meng, Jianwei Yang, Ning Zhang, Xian Li, Ashish Shah, and 11 others. 2026. Agent Learning via Early Experience. In *ICML*.

## A Related Work

### A.1 Training Tool-integrated Agentic Systems

A substantial body of prior work aims to improve the performance of tool-integrated agentic systems under static tool environments. Most approaches train an LLM agent to generate valid tool calling sequences using supervised fine-tuning (SFT) with expert demonstrations or reinforcement learning (RL).

Zhang et al. (2026) propose two SFT strategies that incorporate additional trajectories generated by the agent itself. The first performs “implicit world modeling” to learn transition dynamics from agent–environment interactions before fine-tuning on expert trajectories. The second constructs “self-reflection data” by contrasting agent-generated trajectories with expert demonstrations, encouraging the model to internalize why expert solutions are preferable. Similarly, Qian et al. (2025) design reward functions that align generated trajectories with expert demonstrations, supervising both intermediate tool calls and final outcomes.

In parallel, a complementary line of work develops reinforcement learning methods tailored to tool-integrated agents (Jin et al., 2025; Mai et al., 2025; Xue et al., 2026; Jiang et al., 2025; Mo et al., 2025; Cheng et al., 2025). A key distinction from conventional RL is the masking of feedback tokens from the tool environment, ensuring that parameter updates depend solely on agent-generated tokens. Additional refinements include filtering trajectories containing invalid or incomplete tool calls (Xue et al., 2026) and introducing multi-turn optimization strategies, such as turn-level ratios in group relative policy optimization (GRPO). Beyond individual algorithms, prior studies examine the interaction between SFT and RL for improving generalization (Chu et al., 2025): SFT followed by RL can enhance training stability, while RL-only approaches have been reported to achieve stronger generalization in certain tool-integrated reasoning tasks (Xue et al., 2026).

Despite their methodological differences, these approaches share two assumptions: that the tool environment is static throughout training and evaluation, and that the harness is fixed and treated as part of the environment rather than as a design variable. As a consequence, neither the harness’s role in shaping what the post-trained agent learns, nor the agent’s robustness to changes in the harness or the tool environment, is systematically examined.

### A.2 Harness Design for Tool-integrated Agentic Systems

The harness, as introduced in §1, is widely recognized as a substantial determinant of agent performance, with recent work reporting accuracy gaps of up to roughly six-fold across harnesses (Lee et al., 2026). For tool-integrated agentic systems, the harness specifies, among other things, which tools are exposed to the agent and how they are described, what auxiliary information accompanies each per-step observation, and how the resulting context is assembled into a prompt.

In practice, harnesses have been hand-designed for closed-source models. Benchmarks for interactive LLM agents, including those built on ALFWorld, typically report results obtained under carefully crafted harnesses, often containing detailed descriptions of available actions, structured per-step observations, and various forms of auxiliary information (Zhang et al., 2026). A recent line of work seeks to remove the human from this loop: Lee et al. (2026) introduce Meta-Harness, which casts harness design as a search problem and uses a high-capacity LLM agent to read candidate harnesses and propose new ones automatically. Even with such automated search and strong base models, however, the gains from harness design alone are not always satisfactory. Furthermore, the underlying LLM is held fixed throughout, with the implications for harness design in the post-training regime remaining unaddressed.

In this work, we study how harness design and post-training jointly shape the resulting agent: we treat the harness as an explicit, controlled design dimension and post-train independently under harnesses providing three different levels of auxiliary information, characterizing their interplay both in-distribution and under task and tool environment shift.

### A.3 Tool-integrated Agentic Systems under Dynamic Environments

Chen et al. (2025) introduce ToolQA-D, a benchmark in which API names, signatures, and behaviors drift over time with the agent unaware of these changes, and propose ToolEVO, a framework that uses Monte Carlo Tree Search to enable an agent to detect and adapt to such variability at inference time. More recently, Li et al. (2026) propose ProEvo, a graph-based framework in which the data, tools, and schema of an agent environment are rep-

resented as a typed relational graph and evolved through programmable graph transformations, allowing controlled evaluation of agents' robustness as environments change. While these efforts share our motivation, they neither treat the harness as a design dimension nor consider the post-training regime, leaving the interplay between the two under such shifts uncharacterized.

## B Benchmark Details

### B.1 Harness Configurations

| Tool      | h-low / h-mid                          | h-high                                                                                                                                                                                                                                                                                                      |
|-----------|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Go        | Move to a receptacle.                  | Move to a receptacle in the room. This sets your current location to that receptacle; you must be at a receptacle before you can Take from it, Open/Close it, Examine it, or Move an object to it.                                                                                                          |
| Open      | Open a receptacle.                     | Open a closed container at your current location (drawer, cabinet, fridge, microwave). You must be at the container (use Go) and it must currently be closed. After opening, its contents become visible and you can Take items from it.                                                                    |
| Close     | Close a receptacle.                    | Close an open container at your current location. Normally not required for task completion; useful only when a goal explicitly requires closed state.                                                                                                                                                      |
| Take      | Take an object from a receptacle.      | Pick up an object from a receptacle at your current location. Preconditions: you must be at the receptacle (use Go), the object must be visible, and if the receptacle is a closed container (drawer/cabinet/fridge/microwave) you must Open it first. You can only carry one object at a time.             |
| Move      | Place an object in or on a receptacle. | Place the object you are carrying in/on a receptacle at your current location. Preconditions: you must be carrying the object and be at the target receptacle (for closed containers, Open it first). This action COMPLETES the task when the carried object and the target receptacle both match the goal. |
| Clean     | Clean an object using a receptacle.    | Clean an object you are carrying using a sinkbasin at your current location. Preconditions: you must be carrying the object and be at the sinkbasin.                                                                                                                                                        |
| Heat      | Heat an object using a receptacle.     | Heat an object you are carrying using a microwave at your current location. Preconditions: you must be carrying the object and be at the microwave.                                                                                                                                                         |
| Cool      | Cool an object using a receptacle.     | Cool an object you are carrying using a fridge at your current location. Preconditions: you must be carrying the object and be at the fridge.                                                                                                                                                               |
| Slice     | Slice an object using a sharp object.  | Slice an object at your current location using a knife (or butterknife) you are carrying. Preconditions: you must be carrying the sharp object and be at the target object's location.                                                                                                                      |
| Look      | Look around your current location.     | Look around your current surroundings. If not at any receptacle, reports all visible receptacles in the room. If at a receptacle, reports what is on/in it.                                                                                                                                                 |
| Inventory | Check your current inventory.          | Report what object you are currently carrying. You can carry at most one object at a time.                                                                                                                                                                                                                  |
| Examine   | Examine a receptacle or an object.     | Examine a receptacle or object at your current location. For containers this reveals their contents; for objects this reports their state. Useful for re-checking an already-opened receptacle without Go-ing again.                                                                                        |
| Use       | Use an object.                         | Activate an object at your current location, most commonly turning on a lamp (desk lamp / floor lamp). In 'look at object in light' tasks, turning on the lamp while holding the target object completes the task.                                                                                          |

Table 5: One-line versus rich tool descriptions in *p*, for all 13 tools (under tool schema v1.0). The one-line form is shared by h-low and h-mid; the rich form is used only by h-high. Rows are grouped according to the v2.0 consolidation (Table 9).

| Harness Ver. | Per-step history $\mathcal{T}_t$                                                                                                                                                                         |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| h-low        | Your task is to: <GOAL>.<br>Current location: <LOC><br>You are now at step <N> and your current observation is: <STEP_OBS>                                                                               |
| h-mid        | Your task is to: <GOAL>.<br>Current location: <LOC><br>You are now at step <N> and your current observation is: <STEP_OBS><br>Valid tools: <TOOL_A>, <TOOL_B>, <TOOL_C>                                  |
| h-high       | Your task is to: <GOAL>.<br>Current location: <LOC><br>Carrying: <ITEM_OR_"nothing"><br>You are now at step <N> and your current observation is: <STEP_OBS><br>Valid tools: <TOOL_A>, <TOOL_B>, <TOOL_C> |

Table 6: Per-step history  $\mathcal{T}_t$  under each harness version. The three harnesses share a common core, with each richer version appending one additional auxiliary line (Valid tools: under h-mid, and additionally Carrying: under h-high).

## B.2 Tool Schema Versions

| Meaning                            | v1.0      | v1.1           |
|------------------------------------|-----------|----------------|
| Move to a receptacle               | Go        | NavigateTo     |
| Open a container                   | Open      | OpenContainer  |
| Close a container                  | Close     | CloseContainer |
| Pick up an object                  | Take      | Pickup         |
| Place an object in/on a receptacle | Move      | Place          |
| Wash an object (in sinkbasin)      | Clean     | Wash           |
| Heat an object (in microwave)      | Heat      | Warm           |
| Cool an object (in fridge)         | Cool      | Chill          |
| Slice an object (with a knife)     | Slice     | Cut            |
| Survey the surroundings            | Look      | SurveyRoom     |
| Check items currently carried      | Inventory | CheckCarried   |
| Examine a receptacle or object     | Examine   | Inspect        |
| Activate a device (e.g., lamp)     | Use       | Activate       |

Table 7: Tool name mapping from v1.0 to v1.1, covering all 13 tools defined for ALFWorld. Rows are grouped according to the v2.0 consolidation (Table 9).

| Meaning                                      | v1.0       | v1.1        |
|----------------------------------------------|------------|-------------|
| Destination of a navigation/placement action | receptacle | destination |
| Object being acted upon                      | object     | item        |
| Source receptacle (e.g., for Take)           | receptacle | source      |
| Washing receptacle (sinkbasin)               | receptacle | washer      |
| Heating receptacle (microwave)               | receptacle | heater      |
| Cooling receptacle (fridge)                  | receptacle | cooler      |
| Slicing instrument                           | knife      | blade       |
| Target of an examination                     | target     | subject     |

Table 8: Parameter key mapping from v1.0 to v1.1, covering all parameter roles shared across tools.

| <b>v2.0 Tool</b>  | <b>Merged v1.1 Tools</b>                        | <b>action Values</b>                               | <b>Unified Parameters</b>   |
|-------------------|-------------------------------------------------|----------------------------------------------------|-----------------------------|
| ReceptacleControl | NavigateTo,<br>OpenContainer,<br>CloseContainer | navigate_to,<br>open_container,<br>close_container | action, target              |
| ObjectTransport   | Pickup, Place                                   | pickup, place                                      | action, item,<br>location   |
| ObjectTransform   | Wash, Warm, Chill, Cut                          | wash, warm, chill, cut                             | action, item,<br>instrument |
| Observe           | SurveyRoom,<br>CheckCarried, Inspect            | survey_room,<br>check_carried,<br>inspect          | action, subject             |
| Activate          | Activate                                        | – (single operation)                               | device                      |

Table 9: Tool consolidation from v1.1 (13 tools) to v2.0 (5 tools). Each v2.0 group merges one or more v1.1 tools sharing structural or functional similarity, distinguished within the group by a discrete action parameter. Parameter keys are unified across merged operations.

## C Experimental Details

**Multi-step Tool Calling Process as Markov Decision Process.** We illustrate in detail how multi-step tool calling process is modeled as a Markov decision process (MDP) at a sequence level. Specifically, the MDP is defined by a tuple  $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \mathcal{P}_0)$  as the following.

- $\mathcal{S} := \mathcal{V}^*$ : State space
- $\mathcal{A} := \mathcal{V}^*$ : Action space
- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ : State transition probability such that  $\sum_{s' \in \mathcal{S}} \mathcal{P}(s, a, s') = 1$  for all  $s \in \mathcal{S}$  and  $a \in \mathcal{A}$ . In addition, we let  $\mathcal{P}(s'|s, a) := \mathcal{P}(s, a, s')$ .
- $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ : Step-level reward function evaluated at each step, where we let

$$r_t := r(s_t, a_t, s_{t+1}).$$

- $\mathcal{P}_0 : \mathcal{S} \rightarrow [0, 1]$ : Initial state probability distribution

Then, the multi-step tool calling process can be formalized as follows. Before the process starts, the initial state  $s_1 := (p, q)$  is sampled from  $\mathcal{P}_0$ , where  $p \in \mathcal{V}^*$  is a system prompt and  $q \in \mathcal{V}^*$  is a user query. In each tool calling step  $t$ , the LLM agent  $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$  first observes the state  $s_t$ . Specifically,  $s_t := (p, q, \mathcal{T}_{t-1}) \in \mathcal{S}$ , where  $\mathcal{T}_{t-1}$  is a history of tool calling steps up to the  $(t-1)$ -th step (possibly in a summarized version), and  $\mathcal{T}_0$  is an empty sequence. The agent then generates an action sequence  $a_t \in \mathcal{A}$ . The environment then samples the next state  $s_{t+1}$  from the probability distribution  $\mathcal{P}(\cdot|s_t, a_t)$ , where

$$\mathcal{P}(s_{t+1}|s_t, a_t) := \begin{cases} 1 & s_{t+1} = (p, q, \mathcal{T}_t), \\ 0 & \text{o.w.} \end{cases}$$

Here,  $e_t = \text{TE}((s_t, a_t))$  and  $\mathcal{T}_t = \text{TE}((\mathcal{T}_{t-1}, a_t, e_t))$ . Note that the transition dynamics is deterministic and TE-dependent. In addition, the agent receives the step level reward  $r_t$ . The tool calling process proceeds for  $T \in \mathbb{N}$  steps, where the process terminates either when it succeeds in a given task or when the agent reaches the maximum number of tool calling steps. Finally, by letting  $\tau := \{(s_1, a_1, s_2), \dots, (s_T, a_T, s_{T+1})\} \subset (\mathcal{S} \times \mathcal{A} \times \mathcal{S})^*$  be an episode-level trajectory and

$\mathcal{R} : (\mathcal{S} \times \mathcal{A} \times \mathcal{S})^* \rightarrow \mathbb{R}$  be an episode-level reward, the objective can be formulated as

$$\mathcal{J}(\theta) = \mathbb{E}_{\mathcal{P}_0, \mathcal{P}, \pi_\theta} [\mathcal{R}(\tau)],$$

where  $\mathcal{R}(\tau) := \sum_{t=1}^T \gamma^{t-1} r_t$  is a typical choice ( $\gamma \in [0, 1]$ ).

**Group-relative policy optimization (GRPO)** (Shao et al., 2024). It is a widely adopted algorithm for post-training LLMs due to its computational efficiency in calculating the gradient  $\nabla_\theta \mathcal{J}(\theta)$ . Given a system prompt and a question pair  $(p, q)$ , a behavior policy  $\pi_{\theta_{\text{old}}}$  samples a group of  $G$  episodes  $\{\tau^i\}_{i=1}^G$ , where  $\tau^i := \{(s_1^i, a_1^i, s_2^i), \dots, (s_{T^i}^i, a_{T^i}^i, s_{T^i+1}^i)\}$  is the  $i$ -th episode. Here, we let  $a_t^i := (a_{t1}^i, \dots, a_{tn_t^i}^i)$ , where  $n_t^i := |a_t^i|$ . GRPO algorithm then optimizes the following token-level objective:

$$\mathcal{J}(\theta) = \mathbb{E}_{\substack{(p,q) \sim \mathcal{P}_0, \\ \{a_t^i\} \sim \pi_{\theta_{\text{old}}}}} \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{\sum_{t=1}^{T^i} n_t^i} \sum_{t=1}^{T^i} \sum_{k=1}^{n_t^i} \left\{ \min \left( \rho_{tk}^i \hat{A}^i, \text{clip}(\rho_{tk}^i, 1 - \varepsilon, 1 + \varepsilon) \hat{A}^i \right) - \beta D_{\text{KL}}(\pi_\theta(\cdot | s_{<tk}^i) \| \pi_{\text{ref}}(\cdot | s_{<tk}^i)) \right\} \right].$$

where  $s_{<tk}^i := (p, q, \mathcal{T}_{t-1}^i, a_{t1}^i, \dots, a_{t(k-1)}^i)$  denotes the context preceding token  $a_{tk}^i$ , and the token-level ratio  $\rho_{tk}^i$  and the episode-level advantage  $\hat{A}^i$  are defined as follows:

$$\rho_{tk}^i = \frac{\pi_\theta(a_{tk}^i | s_{<tk}^i)}{\pi_{\theta_{\text{old}}}(a_{tk}^i | s_{<tk}^i)}, \quad \hat{A}^i = \frac{R^i - \mu}{\sigma}.$$

Here,  $\{R^i\}_{i=1}^G$  are episode-level rewards evaluated on  $\{\tau^i\}_{i=1}^G$ , and  $\mu = \text{mean}(\{R^i\}_{i=1}^G)$ ,  $\sigma = \text{std}(\{R^i\}_{i=1}^G)$  are their sample mean and standard deviation, respectively. Here,  $\pi_{\text{ref}}$  is a fixed reference policy, typically initialized from the initial policy and kept frozen throughout training. In practice, the per-token KL term is computed using the low-variance estimator of  $D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$ , as adopted in GRPO (Shao et al., 2024).

**Group-in-group policy optimization (GiGPO)** (Feng et al., 2025). Unlike GRPO, GiGPO differentiates the advantage assigned to each step within the same episode by additionally considering a step-level advantage  $\hat{A}_t^i$  in addition to an episode-level advantage  $\hat{A}^i$ . To define a step-level

advantage, GiGPO algorithm requires a state identifier  $h : \mathcal{V}^* \times \mathcal{V}^* \rightarrow \{0, 1\}$  that verifies whether two different per-step initial states,  $s_{t_1}^i$  and  $s_{t_2}^j$ , i.e., the states from which steps  $t_1$  and  $t_2$  begin, are semantically equivalent (possibly from different episodes, where  $i \neq j$ ). Specifically, given a set of step-level transitions across all  $G$  episodes  $\mathcal{G} := \{(s_t^i, a_t^i, s_{t+1}^i)\}_{i \in [G], t \in [T^i]}$ , GiGPO partitions  $\mathcal{G} = \cup_{\tilde{s} \in \mathcal{AN}} \mathcal{G}(\tilde{s})$  into disjoint groups, where steps within each group  $\mathcal{G}(\tilde{s})$  shares the same per-step initial state  $\tilde{s} \in \mathcal{V}^*$  w.r.t. the state identifier  $h$  as follows:

$$\mathcal{G}(\tilde{s}) := \{(s_t^i, a_t^i, s_{t+1}^i) \in \mathcal{G} : h(s_t^i, \tilde{s}) = 1\}.$$

Here,  $\mathcal{AN}$  is an anchor set that partitions  $\mathcal{G}$  based on the semantic equivalence of the per-step initial state that jointly satisfies

$$\begin{aligned} \mathcal{G} &= \cup_{\tilde{s} \in \mathcal{AN}} \mathcal{G}(\tilde{s}), \\ \mathcal{G}(\tilde{s}) \cap \mathcal{G}(\hat{s}) &= \emptyset \quad \forall \tilde{s}, \hat{s} \in \mathcal{AN} \text{ s.t. } h(\tilde{s}, \hat{s}) = 0. \end{aligned}$$

Then, every step level transition  $(s_t^i, a_t^i, s_{t+1}^i)$  is assigned an index  $g_t^i$  that specifies which group that it belongs to, defined as  $g_t^i := (\tilde{s} \in \mathcal{AN} \text{ s.t. } h(s_t^i, \tilde{s}) = 1)$ . GiGPO algorithm then optimizes the following token-level objective:

$$\begin{aligned} \mathcal{J}(\theta) = \mathbb{E}_{\substack{(p,q) \sim P_0, \\ \{a_t^i\} \sim \pi_{\theta_{\text{old}}}}} & \left[ \frac{1}{G} \sum_{i=1}^G \frac{1}{\sum_{t=1}^{T^i} n_t^i} \sum_{t=1}^{T^i} \sum_{k=1}^{n_t^i} \right. \\ & \left\{ \min \left( \rho_{tk}^i (\hat{A}^i + \omega \hat{A}_t^i), \right. \right. \\ & \quad \left. \text{clip}(\rho_{tk}^i, 1 - \varepsilon, 1 + \varepsilon) (\hat{A}^i + \omega \hat{A}_t^i) \right. \\ & \quad \left. \left. - \beta D_{\text{KL}}(\pi_{\theta}(\cdot | s_{<tk}^i) \| \pi_{\text{ref}}(\cdot | s_{<tk}^i)) \right) \right\}. \end{aligned}$$

where the token level ratio  $\rho_{tk}^i$  and the episode-level advantage  $\hat{A}^i$  are defined in a same manner as GRPO, and the step-level advantage  $\hat{A}_t^i$  is defined as :

$$\hat{A}_t^i = \frac{R_t^i - \mu_{g_t^i}}{\sigma_{g_t^i}}.$$

Here,  $R_t^i := \sum_{\tilde{t}=t}^{T^i} \gamma^{\tilde{t}-t} r_{\tilde{t}}^i$  is a step-level return of the  $t$ -th step within the  $i$ -th episode. For each anchor  $\tilde{s} \in \mathcal{AN}$ ,  $\mu_{\tilde{s}}$  and  $\sigma_{\tilde{s}}$  denote the sample mean and standard deviation of the step-level returns  $\{R_t^i\}$  within  $\mathcal{G}(\tilde{s})$ . The advantage  $\hat{A}_t^i$  then uses the statistics of the group  $g_t^i$  to which the  $t$ -th step of the  $i$ -th episode belongs. Note that the discount

factor  $\gamma \in [0, 1]$  may be different from that used to define the episode-level reward  $R^i$ .  $\omega \in \mathbb{R}^+$  is a hyperparameter that balances episode-level and step-level advantages.

**Training.** We fine-tune Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct with GRPO and GiGPO, using group-normalized advantages over  $G=8$  trajectories per prompt. We train for 500 steps with a constant learning rate of  $1 \times 10^{-6}$ , a batch of 16 task prompts per step, a PPO minibatch size of 256, and a low-variance KL penalty (coefficient 0.01) applied to the loss. A randomly sampled 10% of the training data is held out for validation and used to select the final checkpoint. Rollouts use vLLM with a sampling temperature of 1.0 and a maximum of 50 turns per episode over the 13 ALFWorld tools, with a maximum prompt length of 3072 tokens and a maximum response length of 1024 tokens. The reward is sparse: a successful trajectory receives 10, while each invalid action incurs a penalty of  $-0.1$ . We use  $4 \times$  H200 GPUs for training.

**Evaluation.** We evaluate on the ALFWorld test split of 274 games (140 seen, 134 unseen), with the evaluation harness matched to the training harness and a sampling temperature of 0.4. Each configuration is run with three seeds, and we report the mean success rate; the corresponding per-configuration standard deviations are reported alongside the means in all result tables. For GPT-5-mini, we use a reasoning effort of high; since reasoning models require substantially more tokens, we set the maximum response length to 4096 tokens. Due to budget constraints, GPT-5-mini is run with a single seed. For the tool environment shift scenario, we refer to the paraphrased schema v1.1 as the *mild* shift and the consolidated schema v2.0 as the *strong* shift, reflecting the increasing degree of surface-level deviation from the v1.0 training schema.

## D Additional Experimental Results

| Model Type    | Model               | Test-time Harness | $\mathcal{D}_{\text{easy}}^{\text{te}}$ |                | $\mathcal{D}_{\text{med}}^{\text{te}}$ |                |                | $\mathcal{D}_{\text{hard}}^{\text{te}}$ | All                    |
|---------------|---------------------|-------------------|-----------------------------------------|----------------|----------------------------------------|----------------|----------------|-----------------------------------------|------------------------|
|               |                     |                   | Pick                                    | Look           | Clean                                  | Heat           | Cool           | Pick 2                                  |                        |
| Closed-source | GPT-5 Mini (high)   | h-low             | 71.2                                    | 16.1           | 19.0                                   | 12.8           | 15.2           | 17.1                                    | 28.1                   |
|               |                     | h-mid             | 67.8                                    | 12.9           | 20.7                                   | 20.5           | 34.8           | 12.2                                    | 31.0 (+2.9)            |
|               |                     | h-high            | 93.2                                    | 77.4           | 36.2                                   | 66.7           | 78.3           | 61.0                                    | 68.3 (+40.2)           |
| Open-source   | Qwen2.5-3B-Instruct | h-low             | 6.2 $\pm$ 2.6                           | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                          | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                           | 1.3 $\pm$ 0.6          |
|               |                     | h-mid             | 11.9 $\pm$ 4.5                          | 2.2 $\pm$ 1.9  | 0.0 $\pm$ 0.0                          | 3.4 $\pm$ 3.0  | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                           | 3.3 $\pm$ 1.1 (+2.0)   |
|               |                     | h-high            | 35.0 $\pm$ 2.6                          | 15.1 $\pm$ 1.9 | 11.5 $\pm$ 1.0                         | 12.8 $\pm$ 0.0 | 2.2 $\pm$ 2.2  | 0.0 $\pm$ 0.0                           | 13.9 $\pm$ 0.7 (+12.6) |
|               | Qwen2.5-7B-Instruct | h-low             | 19.8 $\pm$ 3.5                          | 18.3 $\pm$ 4.9 | 0.6 $\pm$ 1.0                          | 3.4 $\pm$ 3.0  | 2.9 $\pm$ 3.3  | 0.0 $\pm$ 0.0                           | 7.4 $\pm$ 0.9          |
|               |                     | h-mid             | 44.6 $\pm$ 2.0                          | 20.4 $\pm$ 3.7 | 9.8 $\pm$ 3.6                          | 8.5 $\pm$ 3.9  | 5.1 $\pm$ 1.3  | 0.0 $\pm$ 0.0                           | 16.1 $\pm$ 1.7 (+8.7)  |
|               |                     | h-high            | 54.8 $\pm$ 3.9                          | 18.3 $\pm$ 6.7 | 26.4 $\pm$ 2.6                         | 35.9 $\pm$ 0.0 | 24.6 $\pm$ 3.3 | 1.6 $\pm$ 1.4                           | 29.0 $\pm$ 1.8 (+21.6) |

Table 10: Zero-shot success rates without post-training: Pretrained agents are evaluated on  $\mathcal{D}_{\text{all}}^{\text{te}}$  under each harness with tool schema fixed to v1.0. GPT-5 Mini is run only once due to the limited budget.

| Model               | Algorithm | Harness | $\mathcal{D}_{\text{easy}}^{\text{te}}$ |                | $\mathcal{D}_{\text{med}}^{\text{te}}$ |                |                | $\mathcal{D}_{\text{hard}}^{\text{te}}$ | All            |
|---------------------|-----------|---------|-----------------------------------------|----------------|----------------------------------------|----------------|----------------|-----------------------------------------|----------------|
|                     |           |         | Pick                                    | Look           | Clean                                  | Heat           | Cool           | Pick 2                                  |                |
| Qwen2.5-3B-Instruct | GRPO      | h-low   | 80.2 $\pm$ 2.6                          | 38.7 $\pm$ 3.2 | 66.1 $\pm$ 3.6                         | 64.1 $\pm$ 0.0 | 66.7 $\pm$ 1.3 | 0.0 $\pm$ 0.0                           | 56.0 $\pm$ 1.1 |
|                     |           | h-mid   | 80.2 $\pm$ 2.6                          | 60.2 $\pm$ 4.9 | 67.2 $\pm$ 1.7                         | 70.9 $\pm$ 1.5 | 68.8 $\pm$ 4.5 | 11.4 $\pm$ 3.7                          | 61.7 $\pm$ 1.0 |
|                     |           | h-high  | 81.4 $\pm$ 1.7                          | 77.4 $\pm$ 5.6 | 75.9 $\pm$ 3.0                         | 71.8 $\pm$ 2.6 | 83.3 $\pm$ 1.3 | 21.1 $\pm$ 3.7                          | 69.7 $\pm$ 0.6 |
|                     | GiGPO     | h-low   | 91.5 $\pm$ 0.0                          | 89.2 $\pm$ 6.7 | 72.4 $\pm$ 7.5                         | 67.5 $\pm$ 3.0 | 64.5 $\pm$ 1.3 | 0.0 $\pm$ 0.0                           | 65.6 $\pm$ 1.3 |
|                     |           | h-mid   | 94.9 $\pm$ 1.7                          | 63.4 $\pm$ 4.9 | 93.7 $\pm$ 2.6                         | 85.5 $\pm$ 3.0 | 82.6 $\pm$ 2.2 | 65.0 $\pm$ 6.1                          | 83.2 $\pm$ 1.3 |
|                     |           | h-high  | 85.9 $\pm$ 5.4                          | 88.2 $\pm$ 1.9 | 89.7 $\pm$ 1.7                         | 84.6 $\pm$ 5.1 | 79.7 $\pm$ 5.5 | 61.8 $\pm$ 1.4                          | 82.1 $\pm$ 2.4 |
| Qwen2.5-7B-Instruct | GRPO      | h-low   | 80.8 $\pm$ 2.0                          | 0.0 $\pm$ 0.0  | 56.9 $\pm$ 1.7                         | 75.2 $\pm$ 3.0 | 52.9 $\pm$ 1.3 | 43.9 $\pm$ 2.4                          | 55.6 $\pm$ 0.8 |
|                     |           | h-mid   | 79.7 $\pm$ 1.7                          | 86.0 $\pm$ 1.9 | 78.2 $\pm$ 1.0                         | 71.8 $\pm$ 2.6 | 66.7 $\pm$ 2.5 | 75.6 $\pm$ 2.4                          | 76.2 $\pm$ 0.2 |
|                     |           | h-high  | 82.5 $\pm$ 2.0                          | 86.0 $\pm$ 1.9 | 81.6 $\pm$ 1.0                         | 75.2 $\pm$ 1.5 | 79.7 $\pm$ 4.5 | 61.0 $\pm$ 4.2                          | 77.9 $\pm$ 0.8 |
|                     | GiGPO     | h-low   | 86.4 $\pm$ 1.7                          | 75.3 $\pm$ 3.7 | 75.9 $\pm$ 0.0                         | 76.9 $\pm$ 0.0 | 89.9 $\pm$ 1.3 | 78.9 $\pm$ 2.8                          | 81.0 $\pm$ 0.0 |
|                     |           | h-mid   | 90.4 $\pm$ 2.0                          | 80.6 $\pm$ 6.5 | 87.4 $\pm$ 1.0                         | 84.6 $\pm$ 0.0 | 72.5 $\pm$ 1.3 | 78.0 $\pm$ 2.4                          | 83.0 $\pm$ 0.8 |
|                     |           | h-high  | 89.3 $\pm$ 2.0                          | 88.2 $\pm$ 1.9 | 92.5 $\pm$ 1.0                         | 76.1 $\pm$ 1.5 | 86.2 $\pm$ 2.5 | 85.4 $\pm$ 2.4                          | 86.9 $\pm$ 0.4 |

Table 11: In-distribution success rates after post-training: Agents are post-trained on  $\mathcal{D}_{\text{all}}^{\text{tr}}$  under each harness with tool schema fixed to v1.0, and evaluated on  $\mathcal{D}_{\text{all}}^{\text{te}}$  under the same harness and schema.

| Algorithm           | Post-hoc Harness | $\mathcal{D}_{\text{easy}}^{\text{te}}$ |                | $\mathcal{D}_{\text{med}}^{\text{te}}$ |                |                | $\mathcal{D}_{\text{hard}}^{\text{te}}$ | All            |                                 |
|---------------------|------------------|-----------------------------------------|----------------|----------------------------------------|----------------|----------------|-----------------------------------------|----------------|---------------------------------|
|                     |                  | Pick                                    | Look           | Clean                                  | Heat           | Cool           | Pick 2                                  |                |                                 |
| Qwen2.5-3B-Instruct | GRPO             | h-mid                                   | 79.7 $\pm$ 3.4 | 49.5 $\pm$ 9.9                         | 67.2 $\pm$ 3.0 | 63.2 $\pm$ 1.5 | 68.8 $\pm$ 1.3                          | 0.0 $\pm$ 0.0  | 57.5 $\pm$ 1.2 ( <b>-4.2</b> )  |
|                     |                  | h-high                                  | 82.5 $\pm$ 1.0 | 50.5 $\pm$ 3.7                         | 72.4 $\pm$ 1.7 | 68.4 $\pm$ 1.5 | 65.9 $\pm$ 1.3                          | 0.0 $\pm$ 0.0  | 59.6 $\pm$ 0.9 ( <b>-10.1</b> ) |
|                     | GiGPO            | h-mid                                   | 89.8 $\pm$ 1.7 | 95.7 $\pm$ 1.9                         | 77.0 $\pm$ 2.0 | 73.5 $\pm$ 1.5 | 73.9 $\pm$ 2.2                          | 0.0 $\pm$ 0.0  | 69.3 $\pm$ 0.0 ( <b>-13.9</b> ) |
|                     |                  | h-high                                  | 89.3 $\pm$ 2.0 | 93.5 $\pm$ 3.2                         | 73.6 $\pm$ 5.5 | 70.9 $\pm$ 1.5 | 72.5 $\pm$ 1.3                          | 0.0 $\pm$ 0.0  | 67.6 $\pm$ 1.1 ( <b>-14.5</b> ) |
| Qwen2.5-7B-Instruct | GRPO             | h-mid                                   | 77.4 $\pm$ 2.6 | 0.0 $\pm$ 0.0                          | 56.9 $\pm$ 3.0 | 76.9 $\pm$ 4.4 | 56.5 $\pm$ 3.8                          | 42.3 $\pm$ 1.4 | 55.5 $\pm$ 1.1 ( <b>-20.7</b> ) |
|                     |                  | h-high                                  | 80.2 $\pm$ 3.5 | 0.0 $\pm$ 0.0                          | 58.0 $\pm$ 3.6 | 72.6 $\pm$ 1.5 | 55.8 $\pm$ 2.5                          | 40.7 $\pm$ 1.4 | 55.4 $\pm$ 1.2 ( <b>-22.5</b> ) |
|                     | GiGPO            | h-mid                                   | 85.9 $\pm$ 1.0 | 55.9 $\pm$ 7.4                         | 75.3 $\pm$ 1.0 | 82.9 $\pm$ 1.5 | 91.3 $\pm$ 0.0                          | 78.0 $\pm$ 2.4 | 79.6 $\pm$ 1.0 ( <b>-3.4</b> )  |
|                     |                  | h-high                                  | 86.4 $\pm$ 0.0 | 80.6 $\pm$ 5.6                         | 75.9 $\pm$ 0.0 | 79.5 $\pm$ 2.6 | 91.3 $\pm$ 0.0                          | 78.9 $\pm$ 1.4 | 82.2 $\pm$ 1.1 ( <b>-4.7</b> )  |

Table 12: Training-time vs. post-hoc harness application: in-distribution success rates on  $\mathcal{D}_{\text{all}}^{\text{te}}$ . For each harness {h-mid, h-high}, we compare an agent post-trained with that harness in place (*training-time*) against an agent post-trained under h-low with that harness applied only at evaluation (*post-hoc*). The parenthesized value is the change in overall accuracy relative to post-training under the same harness applied throughout. Training-time application consistently outperforms post-hoc application.

|                     |           | v1.0                                    |      |                                        |      |      |                                         |             | v1.1                                    |      |                                        |      |      |                                         |             | v2.0                                    |      |                                        |      |      |                                         |             |
|---------------------|-----------|-----------------------------------------|------|----------------------------------------|------|------|-----------------------------------------|-------------|-----------------------------------------|------|----------------------------------------|------|------|-----------------------------------------|-------------|-----------------------------------------|------|----------------------------------------|------|------|-----------------------------------------|-------------|
|                     | Harness   | $\mathcal{D}_{\text{easy}}^{\text{ic}}$ |      | $\mathcal{D}_{\text{med}}^{\text{ic}}$ |      |      | $\mathcal{D}_{\text{hard}}^{\text{ic}}$ | All         | $\mathcal{D}_{\text{easy}}^{\text{ic}}$ |      | $\mathcal{D}_{\text{med}}^{\text{ic}}$ |      |      | $\mathcal{D}_{\text{hard}}^{\text{ic}}$ | All         | $\mathcal{D}_{\text{easy}}^{\text{ic}}$ |      | $\mathcal{D}_{\text{med}}^{\text{ic}}$ |      |      | $\mathcal{D}_{\text{hard}}^{\text{ic}}$ | All         |
|                     |           | Pick                                    | Look | Clean                                  | Heat | Cool | Pick 2                                  |             | Pick                                    | Look | Clean                                  | Heat | Cool | Pick 2                                  |             | Pick                                    | Look | Clean                                  | Heat | Cool | Pick 2                                  |             |
|                     |           |                                         |      |                                        |      |      |                                         |             |                                         |      |                                        |      |      |                                         |             |                                         |      |                                        |      |      |                                         |             |
| Qwen2.5-7B-Instruct | Zero-shot | h-low                                   | 6.2  | 0.0                                    | 0.0  | 0.0  | 0.0                                     | <u>1.3</u>  | 26.0                                    | 14.0 | 0.0                                    | 0.0  | 0.0  | 0.0                                     | <u>7.2</u>  | 18.6                                    | 10.8 | 0.0                                    | 0.0  | 0.7  | 0.0                                     | <u>5.4</u>  |
|                     |           | h-mid                                   | 11.9 | 2.2                                    | 0.0  | 3.4  | 0.0                                     | <u>3.3</u>  | 46.3                                    | 25.8 | 8.0                                    | 4.3  | 8.0  | 0.0                                     | <u>16.5</u> | 14.1                                    | 18.3 | 0.0                                    | 0.0  | 0.7  | 0.0                                     | <u>5.2</u>  |
|                     |           | h-high                                  | 35.0 | 15.1                                   | 11.5 | 12.8 | 2.2                                     | <u>13.9</u> | 40.7                                    | 20.4 | 10.9                                   | 14.5 | 8.0  | 0.0                                     | <u>16.8</u> | 35.6                                    | 28.0 | 5.7                                    | 7.7  | 8.7  | 0.0                                     | <u>14.6</u> |
|                     | GRPO      | h-low                                   | 80.2 | 38.7                                   | 66.1 | 64.1 | 66.7                                    | <u>56.0</u> | 52.0                                    | 28.0 | 10.9                                   | 14.5 | 5.8  | 0.0                                     | <u>19.7</u> | 17.5                                    | 7.5  | 0.6                                    | 0.9  | 2.2  | 0.0                                     | <u>5.2</u>  |
|                     |           | h-mid                                   | 80.2 | 60.2                                   | 67.2 | 70.9 | 68.8                                    | <u>61.7</u> | 72.3                                    | 47.3 | 56.3                                   | 53.8 | 62.3 | 14.6                                    | <u>53.2</u> | 48.0                                    | 26.9 | 12.1                                   | 20.5 | 26.1 | 7.3                                     | <u>24.3</u> |
|                     |           | h-high                                  | 81.4 | 77.4                                   | 75.9 | 71.8 | 83.3                                    | <u>69.7</u> | 71.2                                    | 63.4 | 70.1                                   | 70.9 | 84.8 | 26.0                                    | <u>65.6</u> | 73.4                                    | 58.1 | 66.1                                   | 51.3 | 79.0 | 26.8                                    | <u>60.9</u> |
|                     | GiGPO     | h-low                                   | 91.5 | 89.2                                   | 72.4 | 67.5 | 64.5                                    | <u>65.6</u> | 87.0                                    | 93.5 | 69.5                                   | 68.4 | 57.2 | 0.0                                     | <u>63.4</u> | 20.9                                    | 90.3 | 27.0                                   | 27.4 | 38.4 | 0.0                                     | <u>30.8</u> |
|                     |           | h-mid                                   | 94.9 | 63.4                                   | 93.7 | 85.5 | 82.6                                    | <u>83.2</u> | 92.7                                    | 45.2 | 87.9                                   | 74.4 | 83.3 | 54.5                                    | <u>76.4</u> | 89.8                                    | 37.6 | 47.7                                   | 35.0 | 63.8 | 47.2                                    | <u>56.4</u> |
|                     |           | h-high                                  | 85.9 | 88.2                                   | 89.7 | 84.6 | 79.7                                    | <u>82.1</u> | 81.9                                    | 83.9 | 82.2                                   | 87.2 | 81.2 | 63.4                                    | <u>80.0</u> | 85.9                                    | 71.0 | 61.5                                   | 64.1 | 79.7 | 60.2                                    | <u>71.0</u> |
| Qwen2.5-7B-Instruct | Zero-shot | h-low                                   | 19.8 | 18.3                                   | 0.6  | 3.4  | 2.9                                     | <u>7.4</u>  | 15.8                                    | 25.8 | 5.7                                    | 0.9  | 1.4  | 0.0                                     | <u>7.9</u>  | 35.6                                    | 24.7 | 5.7                                    | 3.4  | 6.5  | 1.6                                     | <u>13.5</u> |
|                     |           | h-mid                                   | 44.6 | 20.4                                   | 9.8  | 8.5  | 5.1                                     | <u>16.1</u> | 58.8                                    | 22.6 | 35.1                                   | 20.5 | 22.5 | 6.5                                     | <u>30.3</u> | 58.8                                    | 32.3 | 16.1                                   | 18.8 | 17.4 | 8.9                                     | <u>26.6</u> |
|                     |           | h-high                                  | 54.8 | 18.3                                   | 26.4 | 35.9 | 24.6                                    | <u>29.0</u> | 39.5                                    | 16.1 | 27.0                                   | 35.9 | 22.5 | 8.9                                     | <u>26.3</u> | 54.8                                    | 46.2 | 31.6                                   | 35.9 | 26.8 | 16.3                                    | <u>35.8</u> |
|                     | GRPO      | h-low                                   | 80.8 | 0.0                                    | 56.9 | 75.2 | 52.9                                    | <u>55.6</u> | 77.4                                    | 0.0  | 59.2                                   | 63.2 | 55.1 | 45.5                                    | <u>54.3</u> | 2.3                                     | 0.0  | 8.0                                    | 0.9  | 0.7  | 1.6                                     | <u>2.7</u>  |
|                     |           | h-mid                                   | 79.7 | 86.0                                   | 78.2 | 71.8 | 66.7                                    | <u>76.2</u> | 83.6                                    | 79.6 | 87.4                                   | 76.9 | 63.8 | 74.0                                    | <u>78.2</u> | 78.0                                    | 77.4 | 73.0                                   | 29.1 | 52.9 | 64.2                                    | <u>63.6</u> |
|                     |           | h-high                                  | 82.5 | 86.0                                   | 81.6 | 75.2 | 79.7                                    | <u>77.9</u> | 78.0                                    | 83.9 | 73.6                                   | 76.9 | 76.1 | 61.8                                    | <u>74.8</u> | 66.7                                    | 76.3 | 28.7                                   | 48.7 | 51.4 | 56.1                                    | <u>53.0</u> |
|                     | GiGPO     | h-low                                   | 86.4 | 75.3                                   | 75.9 | 76.9 | 89.9                                    | <u>81.0</u> | 87.6                                    | 21.5 | 75.9                                   | 73.5 | 90.6 | 79.7                                    | <u>74.9</u> | 36.2                                    | 0.0  | 61.5                                   | 30.8 | 40.6 | 8.1                                     | <u>33.2</u> |
|                     |           | h-mid                                   | 90.4 | 80.6                                   | 87.4 | 84.6 | 72.5                                    | <u>83.0</u> | 92.7                                    | 82.8 | 87.4                                   | 82.9 | 69.6 | 78.0                                    | <u>83.0</u> | 32.2                                    | 24.7 | 52.3                                   | 52.1 | 41.3 | 17.9                                    | <u>37.8</u> |
|                     |           | h-high                                  | 89.3 | 88.2                                   | 92.5 | 76.1 | 86.2                                    | <u>86.9</u> | 85.9                                    | 79.6 | 92.0                                   | 75.2 | 75.4 | 81.3                                    | <u>82.5</u> | 73.4                                    | 84.9 | 94.8                                   | 22.2 | 66.7 | 65.0                                    | <u>69.6</u> |

Table 13: Post-training success rates under tool environment shift: Agents are post-trained on  $\mathcal{D}_{\text{all}}^{\text{tr}}$  under each harness with tool schema fixed to v1.0, and evaluated on  $\mathcal{D}_{\text{all}}^{\text{ic}}$  under the same harness but with schema v1.1 (mild shift) or v2.0 (stronger shift). Zero-shot results are included for reference, where the **Harness** column refers to the test-time harness.

| Model               | Algorithm | Harness | $\mathcal{D}_{\text{easy}}^{\text{te}}$ |                | $\mathcal{D}_{\text{med}}^{\text{te}}$ |                |                | $\mathcal{D}_{\text{hard}}^{\text{te}}$ | All            |
|---------------------|-----------|---------|-----------------------------------------|----------------|----------------------------------------|----------------|----------------|-----------------------------------------|----------------|
|                     |           |         | Pick                                    | Look           | Clean                                  | Heat           | Cool           | Pick 2                                  |                |
| Qwen2.5-3B-Instruct | GRPO      | h-low   | 74.6 $\pm$ 1.7                          | 54.8 $\pm$ 3.2 | 0.6 $\pm$ 1.0                          | 0.0 $\pm$ 0.0  | 2.9 $\pm$ 1.3  | 1.6 $\pm$ 2.8                           | 23.1 $\pm$ 0.6 |
|                     |           | h-mid   | 84.7 $\pm$ 1.7                          | 76.3 $\pm$ 3.7 | 6.9 $\pm$ 0.0                          | 12.8 $\pm$ 6.8 | 9.4 $\pm$ 5.0  | 13.0 $\pm$ 3.7                          | 33.7 $\pm$ 1.4 |
|                     |           | h-high  | 88.7 $\pm$ 2.6                          | 71.0 $\pm$ 8.5 | 74.7 $\pm$ 5.0                         | 62.4 $\pm$ 5.9 | 66.7 $\pm$ 3.3 | 0.8 $\pm$ 1.4                           | 63.1 $\pm$ 1.7 |
|                     | GiGPO     | h-low   | 89.8 $\pm$ 1.4                          | 88.2 $\pm$ 3.0 | 2.9 $\pm$ 0.8                          | 0.0 $\pm$ 0.0  | 9.4 $\pm$ 4.1  | 38.2 $\pm$ 6.1                          | 37.2 $\pm$ 1.0 |
|                     |           | h-mid   | 92.1 $\pm$ 1.6                          | 90.3 $\pm$ 5.3 | 19.5 $\pm$ 2.9                         | 39.3 $\pm$ 6.7 | 36.2 $\pm$ 3.7 | 4.9 $\pm$ 2.0                           | 46.6 $\pm$ 2.4 |
|                     |           | h-high  | 93.8 $\pm$ 2.1                          | 51.6 $\pm$ 2.6 | 30.5 $\pm$ 2.9                         | 43.6 $\pm$ 2.1 | 50.7 $\pm$ 1.0 | 41.5 $\pm$ 2.0                          | 53.4 $\pm$ 1.0 |
| Qwen2.5-7B-Instruct | GRPO      | h-low   | 64.4 $\pm$ 1.7                          | 38.7 $\pm$ 3.2 | 36.2 $\pm$ 3.4                         | 37.6 $\pm$ 5.3 | 36.2 $\pm$ 1.3 | 12.2 $\pm$ 2.4                          | 39.2 $\pm$ 1.7 |
|                     |           | h-mid   | 77.4 $\pm$ 3.5                          | 81.7 $\pm$ 6.7 | 0.6 $\pm$ 1.0                          | 44.4 $\pm$ 6.5 | 36.2 $\pm$ 4.5 | 28.5 $\pm$ 5.1                          | 42.7 $\pm$ 1.0 |
|                     |           | h-high  | 88.7 $\pm$ 1.0                          | 90.3 $\pm$ 0.0 | 71.3 $\pm$ 2.6                         | 63.2 $\pm$ 7.4 | 58.0 $\pm$ 3.3 | 56.1 $\pm$ 2.4                          | 71.5 $\pm$ 1.6 |
|                     | GiGPO     | h-low   | 90.4 $\pm$ 1.0                          | 86.0 $\pm$ 1.9 | 51.7 $\pm$ 0.0                         | 34.2 $\pm$ 1.5 | 29.0 $\pm$ 5.5 | 30.1 $\pm$ 9.9                          | 54.4 $\pm$ 2.4 |
|                     |           | h-mid   | 89.8 $\pm$ 1.7                          | 90.3 $\pm$ 3.2 | 33.9 $\pm$ 1.0                         | 35.0 $\pm$ 1.5 | 26.1 $\pm$ 4.3 | 61.0 $\pm$ 2.4                          | 55.2 $\pm$ 0.8 |
|                     |           | h-high  | 88.1 $\pm$ 2.9                          | 91.4 $\pm$ 1.9 | 63.2 $\pm$ 2.0                         | 53.0 $\pm$ 5.3 | 65.2 $\pm$ 4.3 | 38.2 $\pm$ 3.7                          | 66.9 $\pm$ 0.8 |

Table 14: Post-training success rates under task shift: Agents are post-trained on  $\mathcal{D}_{\text{easy}}^{\text{tr}}$  under each harness with tool schema fixed to v1.0, and evaluated on  $\mathcal{D}_{\text{all}}^{\text{te}}$  under the same harness and schema.

| Model               | Algorithm | Harness | $\mathcal{D}_{\text{easy}}^{\text{te}}$ |                | $\mathcal{D}_{\text{med}}^{\text{te}}$ |                |                | $\mathcal{D}_{\text{hard}}^{\text{te}}$ | All            |
|---------------------|-----------|---------|-----------------------------------------|----------------|----------------------------------------|----------------|----------------|-----------------------------------------|----------------|
|                     |           |         | Pick                                    | Look           | Clean                                  | Heat           | Cool           | Pick 2                                  |                |
| Qwen2.5-3B-Instruct | GRPO      | h-low   | 6.2 $\pm$ 2.6                           | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                          | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                           | 1.3 $\pm$ 0.6  |
|                     |           | h-mid   | 59.9 $\pm$ 1.0                          | 2.2 $\pm$ 1.9  | 64.4 $\pm$ 5.0                         | 57.3 $\pm$ 1.5 | 36.2 $\pm$ 6.6 | 3.3 $\pm$ 3.7                           | 41.5 $\pm$ 2.4 |
|                     |           | h-high  | 81.9 $\pm$ 1.0                          | 32.3 $\pm$ 3.2 | 81.0 $\pm$ 1.7                         | 89.7 $\pm$ 0.0 | 85.5 $\pm$ 1.3 | 27.6 $\pm$ 7.8                          | 69.7 $\pm$ 1.1 |
|                     | GiGPO     | h-low   | 6.2 $\pm$ 2.6                           | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                          | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                           | 1.3 $\pm$ 0.6  |
|                     |           | h-mid   | 53.1 $\pm$ 1.0                          | 18.3 $\pm$ 4.9 | 81.0 $\pm$ 1.7                         | 76.9 $\pm$ 2.6 | 81.2 $\pm$ 1.3 | 17.9 $\pm$ 3.7                          | 57.9 $\pm$ 1.1 |
|                     |           | h-high  | 51.4 $\pm$ 3.9                          | 12.9 $\pm$ 3.2 | 91.4 $\pm$ 1.7                         | 86.3 $\pm$ 3.0 | 86.2 $\pm$ 2.5 | 10.6 $\pm$ 1.4                          | 60.2 $\pm$ 0.6 |
| Qwen2.5-7B-Instruct | GRPO      | h-low   | 61.0 $\pm$ 1.7                          | 11.8 $\pm$ 1.9 | 75.3 $\pm$ 3.6                         | 73.5 $\pm$ 3.9 | 72.5 $\pm$ 2.5 | 35.0 $\pm$ 5.1                          | 58.3 $\pm$ 2.0 |
|                     |           | h-mid   | 59.3 $\pm$ 4.5                          | 28.0 $\pm$ 6.7 | 64.9 $\pm$ 2.0                         | 70.1 $\pm$ 1.5 | 58.0 $\pm$ 2.5 | 26.0 $\pm$ 2.8                          | 53.3 $\pm$ 1.9 |
|                     |           | h-high  | 86.4 $\pm$ 0.0                          | 88.2 $\pm$ 3.7 | 89.1 $\pm$ 1.0                         | 78.6 $\pm$ 3.9 | 79.7 $\pm$ 1.3 | 34.1 $\pm$ 4.2                          | 77.1 $\pm$ 0.4 |
|                     | GiGPO     | h-low   | 54.8 $\pm$ 2.6                          | 32.3 $\pm$ 3.2 | 73.0 $\pm$ 1.0                         | 75.2 $\pm$ 3.0 | 79.7 $\pm$ 1.3 | 4.9 $\pm$ 2.4                           | 55.7 $\pm$ 0.2 |
|                     |           | h-mid   | 68.4 $\pm$ 3.5                          | 47.3 $\pm$ 8.1 | 71.8 $\pm$ 1.0                         | 76.1 $\pm$ 3.9 | 67.4 $\pm$ 0.0 | 17.1 $\pm$ 2.4                          | 60.0 $\pm$ 0.6 |
|                     |           | h-high  | 81.4 $\pm$ 3.4                          | 23.7 $\pm$ 1.9 | 96.0 $\pm$ 1.0                         | 82.1 $\pm$ 4.4 | 92.0 $\pm$ 3.3 | 52.0 $\pm$ 6.1                          | 75.4 $\pm$ 1.3 |

Table 15: Post-training success rates under task shift: Agents are post-trained on  $\mathcal{D}_{\text{med}}^{\text{tr}}$  under each harness with tool schema fixed to v1.0, and evaluated on  $\mathcal{D}_{\text{all}}^{\text{te}}$  under the same harness and schema. For Qwen2.5-3B-Instruct under h-low, post-training fails to take effect, so zero-shot results are reported instead.

| Model               | Algorithm | Harness | $\mathcal{D}_{\text{easy}}^{\text{te}}$ |                | $\mathcal{D}_{\text{med}}^{\text{te}}$ |                |                | $\mathcal{D}_{\text{hard}}^{\text{te}}$ | All            |
|---------------------|-----------|---------|-----------------------------------------|----------------|----------------------------------------|----------------|----------------|-----------------------------------------|----------------|
|                     |           |         | Pick                                    | Look           | Clean                                  | Heat           | Cool           | Pick 2                                  |                |
| Qwen2.5-3B-Instruct | GRPO      | h-low   | 6.2 $\pm$ 2.6                           | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                          | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                           | 1.3 $\pm$ 0.6  |
|                     |           | h-mid   | 11.9 $\pm$ 4.5                          | 2.2 $\pm$ 1.9  | 0.0 $\pm$ 0.0                          | 3.4 $\pm$ 3.0  | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                           | 3.3 $\pm$ 1.1  |
|                     |           | h-high  | 35.0 $\pm$ 2.6                          | 15.1 $\pm$ 1.9 | 11.5 $\pm$ 1.0                         | 12.8 $\pm$ 0.0 | 2.2 $\pm$ 2.2  | 0.0 $\pm$ 0.0                           | 13.9 $\pm$ 0.7 |
|                     | GiGPO     | h-low   | 6.2 $\pm$ 2.6                           | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                          | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                           | 1.3 $\pm$ 0.6  |
|                     |           | h-mid   | 11.9 $\pm$ 4.5                          | 2.2 $\pm$ 1.9  | 0.0 $\pm$ 0.0                          | 3.4 $\pm$ 3.0  | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                           | 3.3 $\pm$ 1.1  |
|                     |           | h-high  | 35.0 $\pm$ 2.6                          | 15.1 $\pm$ 1.9 | 11.5 $\pm$ 1.0                         | 12.8 $\pm$ 0.0 | 2.2 $\pm$ 2.2  | 0.0 $\pm$ 0.0                           | 13.9 $\pm$ 0.7 |
| Qwen2.5-7B-Instruct | GRPO      | h-low   | 80.8 $\pm$ 1.0                          | 15.1 $\pm$ 9.9 | 13.8 $\pm$ 4.6                         | 0.9 $\pm$ 1.5  | 4.3 $\pm$ 2.2  | 68.3 $\pm$ 4.2                          | 33.1 $\pm$ 0.2 |
|                     |           | h-mid   | 71.8 $\pm$ 1.0                          | 44.1 $\pm$ 1.9 | 13.8 $\pm$ 1.7                         | 6.0 $\pm$ 3.9  | 13.0 $\pm$ 4.3 | 65.0 $\pm$ 2.8                          | 36.1 $\pm$ 0.0 |
|                     |           | h-high  | 81.4 $\pm$ 1.7                          | 53.8 $\pm$ 9.3 | 31.0 $\pm$ 3.0                         | 29.1 $\pm$ 1.5 | 24.6 $\pm$ 7.0 | 81.3 $\pm$ 1.4                          | 50.6 $\pm$ 1.2 |
|                     | GiGPO     | h-low   | 41.2 $\pm$ 1.0                          | 17.2 $\pm$ 4.9 | 15.5 $\pm$ 0.0                         | 3.4 $\pm$ 1.5  | 0.0 $\pm$ 0.0  | 95.1 $\pm$ 0.0                          | 28.8 $\pm$ 0.4 |
|                     |           | h-mid   | 81.4 $\pm$ 1.7                          | 31.2 $\pm$ 1.9 | 7.5 $\pm$ 1.0                          | 18.8 $\pm$ 3.9 | 5.8 $\pm$ 1.3  | 83.7 $\pm$ 3.7                          | 38.8 $\pm$ 0.6 |
|                     |           | h-high  | 95.5 $\pm$ 1.0                          | 93.5 $\pm$ 3.2 | 54.0 $\pm$ 7.0                         | 56.4 $\pm$ 4.4 | 51.4 $\pm$ 5.5 | 94.3 $\pm$ 3.7                          | 73.4 $\pm$ 2.9 |

Table 16: Post-training success rates under task shift: Agents are post-trained on  $\mathcal{D}_{\text{hard}}^{\text{tr}}$  under each harness with tool schema fixed to v1.0, and evaluated on  $\mathcal{D}_{\text{all}}^{\text{te}}$  under the same harness and schema. For Qwen2.5-3B-Instruct, post-training fails to take effect, so zero-shot results are reported instead.

## E Ablation Study

### E.1 Generalization to a Different Agentic Environment

To examine whether our findings extend beyond ALFWorld, we additionally conduct experiments on WebShop (Yao et al., 2022), an interactive e-commerce environment in which an agent must purchase a product satisfying a user request. While ALFWorld focuses on household planning, WebShop requires stateful interaction with an e-commerce interface, providing a complementary setting for evaluating tool-integrated agents. Since WebShop does not provide task-type annotations analogous to the task categories in ALFWorld, we focus on tool environment shift.

**Tool Schema.** We construct two tool schemas while preserving the underlying task and available functionality. Under the base schema v1.0, the agent is provided with a Search tool for retrieving products and a general Click tool for interacting with clickable elements, including selecting product options, viewing product details, purchasing an item, and navigating between pages. Under the shifted schema v2.0, Search remains unchanged, while the general Click tool is decomposed into five specialized tools according to the semantic role of each clickable element. For example, selecting a product option requires ChooseOption rather than the generic Click tool. Thus, the task semantics and functionality remain fixed, while the action interface changes.

**Harness.** We instantiate two WebShop-specific harnesses, h-low and h-high. Under h-low, the agent must infer the currently selected product options from the interaction history. h-high additionally appends the currently selected options (Selected:) to every per-step observation, explicitly exposing the task-relevant interaction state. This plays a role analogous to the Carrying: information provided by h-high in ALFWorld. Tool descriptions are identical across the two harnesses. In addition, because the WebShop page itself exposes clickable elements, both harnesses provide the currently admissible actions: removing this information would introduce artificial partial observability rather than a realistic low-information harness.

**Training and Evaluation Protocol.** Following the tool environment shift protocol in §4.1, we post-train Qwen2.5-3B-Instruct with both GRPO and GiGPO under each harness, with the training tool schema fixed to v1.0. We then evaluate each agent under the same harness using either v1.0 for in-distribution evaluation or v2.0 for tool environment shift. We evaluate on a held-out split of 500 instructions over three seeds. We report the mean WebShop task score, success rate (SR), and the average number of tool-calling turns, with a maximum of 15 turns per episode.

**Results.** The WebShop results reproduce the main trend observed in ALFWorld: post-training with the more informative harness yields stronger performance under tool environment shift (Table 17). Across both GRPO and GiGPO, agents post-trained under h-high achieve higher performance under v2.0 than their h-low counterparts. The effect is most pronounced with GiGPO. Under h-low, the success rate drops from 54.7 to 33.5, whereas under h-high it decreases only from 65.3 to 61.9. The average number of turns also increases substantially more under h-low (5.0  $\rightarrow$  10.6) than under h-high (5.4  $\rightarrow$  7.4). These results indicate that the benefit of harness-aware post-training under a tool environment shift extends beyond ALFWorld to another agentic environment.

| Algorithm | Harness | v1.0           |                |               | v2.0           |                |                |
|-----------|---------|----------------|----------------|---------------|----------------|----------------|----------------|
|           |         | Score          | SR             | Turns         | Score          | SR             | Turns          |
| GRPO      | h-low   | 71.4 $\pm$ 0.5 | 57.5 $\pm$ 1.0 | 6.8 $\pm$ 0.0 | 54.2 $\pm$ 0.7 | 43.4 $\pm$ 1.6 | 10.0 $\pm$ 0.1 |
|           | h-high  | 78.9 $\pm$ 0.2 | 63.7 $\pm$ 1.1 | 6.2 $\pm$ 0.1 | 57.9 $\pm$ 0.5 | 48.7 $\pm$ 0.2 | 9.6 $\pm$ 0.1  |
| GiGPO     | h-low   | 75.6 $\pm$ 0.1 | 54.7 $\pm$ 0.3 | 5.0 $\pm$ 0.0 | 46.9 $\pm$ 0.6 | 33.5 $\pm$ 1.1 | 10.6 $\pm$ 0.1 |
|           | h-high  | 82.1 $\pm$ 0.5 | 65.3 $\pm$ 0.3 | 5.4 $\pm$ 0.1 | 74.3 $\pm$ 0.7 | 61.9 $\pm$ 1.6 | 7.4 $\pm$ 0.1  |

Table 17: Post-training performance of Qwen2.5-3B-Instruct on WebShop under tool environment shift. We report the score, success rate (SR), and the average number of turns across three random seeds.

## E.2 Short Adaptation for Post-hoc Harness Application

We further examine whether a short adaptation stage can close the performance gap of post-hoc harness application. Starting from Qwen2.5-3B-Instruct trained under h-low, we continue post-training under the target harness for 100 additional steps, 20% of the main training budget (Table 18). Across both GRPO and GiGPO, short adaptation improves over naive post-hoc application, but training with the target harness from the beginning remains the strongest. For example, with GiGPO under h-high, adaptation improves success rate from 67.6 to 75.9, while training-time application achieves 82.1. A similar trend holds for GRPO (59.6  $\rightarrow$  63.3 vs. 69.7).

| Algorithm | Harness Application | h-mid                          | h-high                         |
|-----------|---------------------|--------------------------------|--------------------------------|
| GRPO      | Post-hoc (naive)    | 57.5 $\pm$ 1.2                 | 59.6 $\pm$ 0.9                 |
|           | Post-hoc (adapted)  | 58.2 $\pm$ 1.3                 | 63.3 $\pm$ 0.2                 |
|           | Training-time       | <b>61.7<math>\pm</math>1.0</b> | <b>69.7<math>\pm</math>0.6</b> |
| GiGPO     | Post-hoc (naive)    | 69.3 $\pm$ 0.0                 | 67.6 $\pm$ 1.1                 |
|           | Post-hoc (adapted)  | 71.5 $\pm$ 0.6                 | 75.9 $\pm$ 0.4                 |
|           | Training-time       | <b>83.2<math>\pm</math>1.3</b> | <b>82.1<math>\pm</math>2.4</b> |

Table 18: Naive post-hoc, short-adaptation, and training-time harness application: in-distribution success rates on  $\mathcal{D}_{\text{all}}^{\text{te}}$  for Qwen2.5-3B-Instruct post-trained with GRPO or GiGPO. For each target harness {h-mid, h-high}, *Post-hoc (naive)* directly evaluates the agent post-trained under h-low using the target harness; *Post-hoc (adapted)* further adapts the h-low-trained checkpoint to the target harness for 100 steps (20% of the main training steps), selecting the best checkpoint on the validation set; and *Training-time* denotes agents post-trained under the target harness from the beginning.

## E.3 Fine-grained Task Shift Analysis

We further evaluate task shift at the single-category level using Qwen2.5-3B-Instruct, by post-training agents on Pick or Clean only and evaluating on all task categories (Table 19). Across both GRPO and GiGPO, more informative harnesses tend to improve transfer to unseen categories. For example, with GiGPO trained on Pick, h-high improves overall success from 47.4 to 68.5 relative to h-low, including a substantial gain on held-out Pick 2 (6.5  $\rightarrow$  52.8). Although some individual categories exhibit non-monotonic behavior, the aggregate trend aligns with Observation 5: informative harnesses support inter-task transfer even from a single training category.

| Training Split                           | Algorithm | Harness | $\mathcal{D}_{\text{easy}}^{\text{te}}$ |                | $\mathcal{D}_{\text{med}}^{\text{te}}$ |                |                | $\mathcal{D}_{\text{hard}}^{\text{te}}$ | All            |
|------------------------------------------|-----------|---------|-----------------------------------------|----------------|----------------------------------------|----------------|----------------|-----------------------------------------|----------------|
|                                          |           |         | Pick                                    | Look           | Clean                                  | Heat           | Cool           | Pick 2                                  |                |
| $\mathcal{D}_{\text{Pick}}^{\text{tr}}$  | GRPO      | h-low   | 83.1 $\pm$ 1.7                          | 12.9 $\pm$ 3.2 | 3.4 $\pm$ 1.7                          | 3.4 $\pm$ 1.5  | 10.1 $\pm$ 4.5 | 26.8 $\pm$ 6.5                          | 26.3 $\pm$ 0.4 |
|                                          |           | h-mid   | 83.1 $\pm$ 2.9                          | 19.4 $\pm$ 6.5 | 50.6 $\pm$ 1.0                         | 30.8 $\pm$ 2.6 | 26.8 $\pm$ 3.3 | 0.0 $\pm$ 0.0                           | 39.7 $\pm$ 0.2 |
|                                          |           | h-high  | 76.8 $\pm$ 4.3                          | 44.1 $\pm$ 1.9 | 35.1 $\pm$ 2.6                         | 24.8 $\pm$ 5.9 | 21.7 $\pm$ 2.2 | 56.9 $\pm$ 2.8                          | 44.6 $\pm$ 0.4 |
|                                          | GiGPO     | h-low   | 93.8 $\pm$ 2.1                          | 43.0 $\pm$ 3.0 | 51.1 $\pm$ 1.6                         | 27.4 $\pm$ 8.7 | 39.9 $\pm$ 3.7 | 6.5 $\pm$ 2.3                           | 47.4 $\pm$ 0.8 |
|                                          |           | h-mid   | 94.4 $\pm$ 2.9                          | 17.2 $\pm$ 3.0 | 41.4 $\pm$ 6.5                         | 46.2 $\pm$ 3.6 | 31.9 $\pm$ 1.0 | 27.6 $\pm$ 1.1                          | 47.1 $\pm$ 2.1 |
|                                          |           | h-high  | 91.0 $\pm$ 2.1                          | 52.7 $\pm$ 1.5 | 80.5 $\pm$ 3.3                         | 53.8 $\pm$ 6.3 | 61.6 $\pm$ 2.0 | 52.8 $\pm$ 3.0                          | 68.5 $\pm$ 1.5 |
| $\mathcal{D}_{\text{Clean}}^{\text{tr}}$ | GRPO      | h-low   | 6.2 $\pm$ 2.6                           | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                          | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                           | 1.3 $\pm$ 0.6  |
|                                          |           | h-mid   | 61.6 $\pm$ 2.0                          | 28.0 $\pm$ 1.9 | 63.2 $\pm$ 1.0                         | 32.5 $\pm$ 5.3 | 41.3 $\pm$ 3.8 | 2.4 $\pm$ 2.4                           | 41.7 $\pm$ 0.8 |
|                                          |           | h-high  | 83.6 $\pm$ 1.0                          | 28.0 $\pm$ 3.7 | 89.1 $\pm$ 3.6                         | 59.8 $\pm$ 5.3 | 64.5 $\pm$ 5.5 | 0.8 $\pm$ 1.4                           | 59.5 $\pm$ 1.7 |
|                                          | GiGPO     | h-low   | 6.2 $\pm$ 2.6                           | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                          | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0  | 0.0 $\pm$ 0.0                           | 1.3 $\pm$ 0.6  |
|                                          |           | h-mid   | 71.8 $\pm$ 2.6                          | 5.4 $\pm$ 1.9  | 91.4 $\pm$ 1.7                         | 74.4 $\pm$ 6.8 | 77.5 $\pm$ 4.5 | 18.7 $\pm$ 6.1                          | 61.8 $\pm$ 1.1 |
|                                          |           | h-high  | 73.4 $\pm$ 2.1                          | 41.9 $\pm$ 4.6 | 90.8 $\pm$ 1.6                         | 52.1 $\pm$ 3.2 | 74.6 $\pm$ 1.0 | 36.6 $\pm$ 2.0                          | 65.2 $\pm$ 2.0 |

Table 19: Post-training success rates under task shift at the single-category level. Qwen2.5-3B-Instruct agents are post-trained with GRPO or GiGPO on training instances from a single task category (Pick or Clean) under each harness, with tool schema fixed to v1.0, and evaluated on  $\mathcal{D}_{\text{all}}^{\text{te}}$  under the same harness and schema. For the Clean-trained agents under h-low, post-training fails to take effect, so zero-shot results are reported instead.