WORLDPO: HOLISTIC AGENTIC LEARNING WITH POLICY OPTIMIZATION AND WORLD MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Although pretrained large language models possess broad capabilities, limited environment-specific knowledge can hinder their adaptation to unfamiliar agentic environments. Policy optimization improves behavior through task feedback, yet uses information-rich environmental observations primarily as decision context, leaving their value for environmental understanding underexploited. We propose **WorldPO** for *holistic agentic learning* that integrates behavioral improvement with environmental understanding through collaborative policy optimization and world modeling. Using the same on-policy rollouts, WorldPO combines feedback-guided policy learning on generated responses with world modeling supervised by environmental observations, optimizing both objectives in a shared LLM. We evaluate WorldPO in three environments with distinct observations: ALFWorld, WebShop, and Search. WorldPO improves task completion and complements GiGPO's action-level optimization, raising 3B ALFWorld success from 75.00% to 92.19%. On Search, 3B WorldPO outperforms GiGPO on all four non-training datasets. Our experiments reveal that WorldPO benefits from task-relevant environmental knowledge, which is further supported by our analyses of agent behavior.

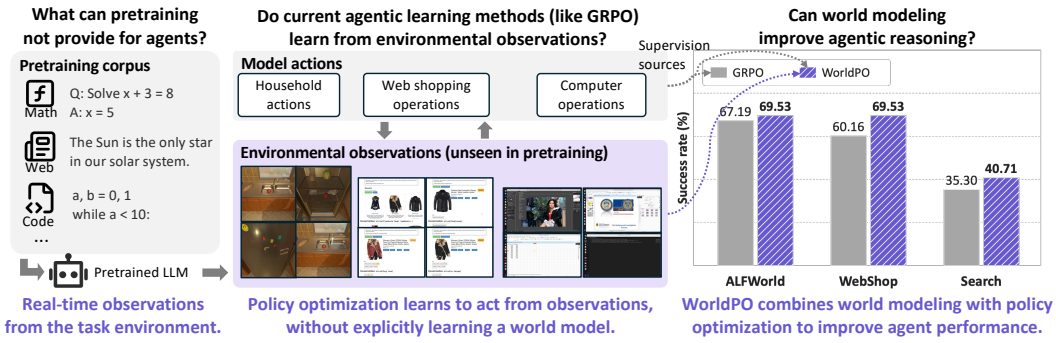

Figure 1: **Motivation for WorldPO.** Environmental observations provide a source of predictive supervision during agent interaction. WorldPO combines observation prediction with policy optimization; the right panel compares 3B model performance on ALFWorld, WebShop, and Search.

1 INTRODUCTION

Large language models (LLMs) achieve strong performance in mathematical reasoning, code generation, and knowledge-intensive question answering (Chowdhery et al., 2023; Yang et al., 2024). However, transferring these capabilities to unfamiliar agentic environments remains challenging: pretraining alone cannot fully cover the dynamic conditions of downstream tasks (Fig. 1), such as operating-system and terminal tasks (Xie et al., 2024; Bui, 2026) or grounding target elements in graphical user interfaces (Qin et al., 2025). Limited environment-specific knowledge can hinder LLMs' understanding of task requirements, available actions, and environmental observations, constraining expert-level performance and reliable agent deployment in multi-turn environments.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

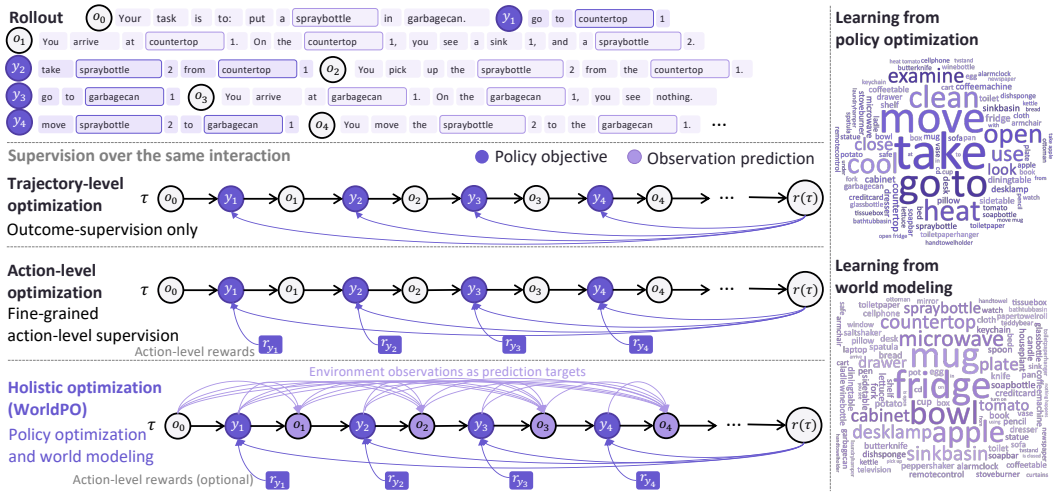

Figure 2: **Schematic illustration of holistic optimization.** A rollout τ contains generated responses y and environmental observations o. Trajectory- and action-level optimization supervise behavior using $r(\tau)$ and r_y, respectively. WorldPO adds world modeling to learn from observations in the same LLM. The word clouds illustrate complementary behavioral and environmental knowledge.

To bridge this gap, policy optimization (*e.g.*, reinforcement learning) uses task feedback to improve agents' task-completion capabilities (Jin et al., 2025; Feng et al., 2025). Environmental observations, however, contain rich task-specific knowledge about environmental states and action consequences. Policy-only methods primarily use these observations to guide behavior, leaving their value for environmental understanding underexploited. Studies show that learning from observations helps agents anticipate action consequences and complete tasks more effectively (Zhang et al., 2025; Guo et al., 2025; Shrivastava et al., 2026). We therefore ask the question: *How can agent learning integrate environmental understanding with behavioral improvement to develop more capable LLM agents?*

To answer this question, we propose **WorldPO** for *holistic agentic learning with policy optimization and world modeling* (Fig. 2). WorldPO enhances agentic capability through a collaborative learning objective combining policy optimization for behavioral improvement with world modeling for environmental understanding. Given on-policy rollouts, WorldPO performs two complementary optimizations: *(1) policy optimization* uses feedback-derived advantages to optimize the policy objective over generated responses; *(2) world modeling* trains the model to predict the next environmental observation conditioned on the agentic history and current generated response. The corresponding losses form a weighted collaborative objective that updates the shared parameters of the same LLM.

We evaluate WorldPO in three environments with distinct observations: household description in ALFWorld, product information in WebShop, and retrieved passages in Search. **(1) Broad effectiveness.** WorldPO improves task completion across environments and policy optimizers. **(2) Complementary optimization.** With GiGPO's action-level optimization, world modeling raises 3B ALFWorld success from 75.00% to 92.19%. **(3) Cross-dataset generalization.** 3B WorldPO exceeds GiGPO on all four unseen Search datasets. Further, we reveal that WorldPO benefits from task-relevant environmental knowledge beyond exact action–observation correspondences, with successful experience providing particularly useful supervision. Our analyses further support this finding: compared with GRPO, WorldPO takes fewer invalid actions and uses less interaction on jointly solved tasks across all three environments, reflecting more reliable and efficient task execution.

In summary, our contributions are threefold:

- **WorldPO for holistic agentic learning.** We propose a framework that collaboratively optimizes policy behavior and observation prediction in the same LLM using on-policy experience (Sec. 3).
- **Evaluation across diverse environments.** Comprehensive experiments across three environments, multiple model scales, and policy optimizers demonstrate improved agent task performance (Sec. 4).
- **Understanding the benefits of world modeling.** We investigate potential sources of improvement through extensive analyses and provide direct evidence of more reliable agent behavior (Sec. 5).

2 PRELIMINARIES

Agentic policy optimization. For a task $x \sim \mathcal{D}$, an agent interacts with an environment $\mathcal{E}$, modeled as a finite-horizon partially observable Markov decision process (POMDP) (Kaelbling et al., 1998). The agent receives environmental observations rather than the full environment state. At turn t, the policy conditions on the agentic history $h_t = (x, o_0, y_0, o_1, \ldots, y_{t-1}, o_t)$, where o_0 is the initial observation, and generates a response $y_t \sim \pi_\theta(\cdot \mid h_t)$ containing reasoning and an executable action a_t. The environment executes a_t extracted from y_t and returns o_{t+1}, such as a tool result or a retrieved webpage, which extends the history together with y_t and conditions the next generated response y_{t+1}.

A completed T-turn trajectory $\tau = (x, o_0, y_0, \ldots, y_{T-1}, o_T)$ receives a task return $r(\tau)$ and follows the distribution $p_{\theta,\mathcal{E}}(\tau \mid x)$ induced by the policy and environment. Policy optimization maximizes the expected return. For a fixed environment and return function, the objective and its gradient are

$$J(\theta) = \mathbb{E}_{x,\tau}[r(\tau)], \qquad \nabla_\theta J(\theta) = \mathbb{E}_{x,\tau}\left[r(\tau) \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(y_t \mid h_t)\right]. \tag{1}$$

Both expectations are over $x \sim \mathcal{D}$ and $\tau \sim p_{\theta,\mathcal{E}}(\cdot \mid x)$. The return weights each generated response's log-probability gradient, while observations condition the policy through the agentic history. Practical methods such as GRPO (Shao et al., 2024), RLOO (Ahmadian et al., 2024), and GiGPO (Feng et al., 2025) use advantage estimates and surrogate objectives to assign credit to generated responses. We denote the selected policy surrogate loss, including its configured regularization terms, by $\mathcal{L}_{\mathrm{PO}}(\theta)$.

Agentic world modeling. World models learn to predict how an environment evolves under an agent's actions (Hafner et al., 2019). In agentic interactions, we consider next-observation prediction with a predictor $p_\phi(o_{t+1} \mid h_t, y_t)$. The agentic history and current generated response form the input; the next environmental observation provides supervision for predicting the observed consequence of the executed action. The prediction task admits different training objectives. For example, supervised fine-tuning (SFT) maximizes the likelihood of environmental observations, while direct preference optimization (DPO) can use preferences over candidate observation predictions (Wang et al., 2025b). We denote the selected world-modeling loss by $\mathcal{L}_{\mathrm{WM}}(\phi)$. Appendix H provides the trajectory probability decomposition, policy-gradient derivation, and likelihood-based prediction objective.

3 WORLDPO: HOLISTIC AGENTIC LEARNING

As illustrated in Fig. 3, WorldPO couples behavioral improvement with environmental understanding through policy optimization and auxiliary world modeling in a shared LLM with parameters θ. The same on-policy trajectories provide outcome feedback for policy learning and environmental observations for prediction, allowing these two forms of supervision to update shared parameters.

3.1 WORLD MODELING FROM AGENTIC EXPERIENCE

A task return evaluates the outcome of a trajectory, whereas environmental observations provide task-specific knowledge about environmental states and action consequences. For example, after an ALFWorld agent successfully takes a pencil from a desk, the environment returns an observation confirming the pickup. Even if the agent later fails to place the pencil at the required destination, this observation remains a prediction target for the completed action. WorldPO therefore uses the same on-policy experience to learn which generated responses lead to successful outcomes and to predict subsequent observations. We instantiate the latter through likelihood fitting of o_{t+1}, conditioned on the agentic history h_t and current generated response y_t, using the returned observations as targets.

Let $\mathcal{B}$ contain the collected transitions (h_t, y_t, o_{t+1}), with $N_{\mathcal{B}} = \sum_{(h_t,y_t,o_{t+1})\in\mathcal{B}} |o_{t+1}| > 0$ retained observation tokens in total. Writing $o_{t+1,j}$ for token j and $o_{t+1,<j}$ for its preceding tokens, the observation-prediction loss averages negative log-likelihood (NLL) over retained observation tokens:

$$\mathcal{L}_{\mathrm{WM}}(\theta; \mathcal{B}) = -\frac{1}{N_{\mathcal{B}}} \sum_{(h_t,y_t,o_{t+1})\in\mathcal{B}} \sum_{j=1}^{|o_{t+1}|} \log p_\theta(o_{t+1,j} \mid h_t, y_t, o_{t+1,<j}). \tag{2}$$

Here, h_t, y_t, and o_{t+1} denote the token sequences retained after preprocessing and truncation. History and generated response tokens provide conditioning information but receive no observation-prediction

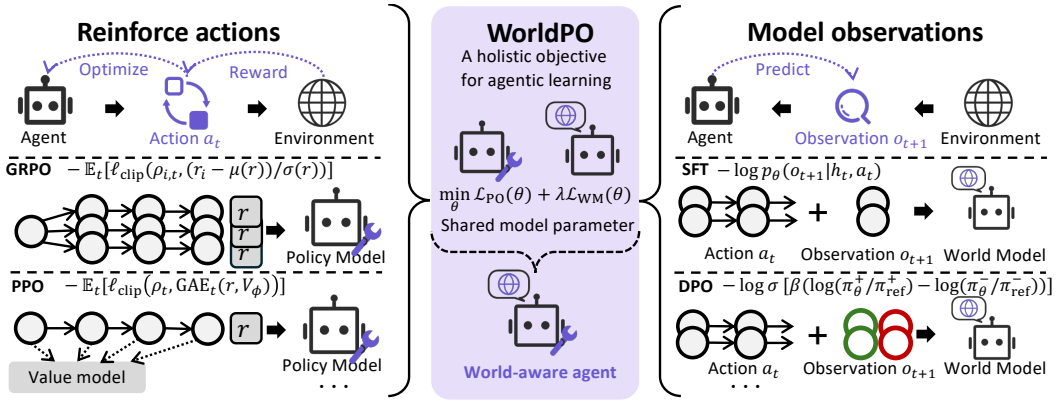

Figure 3: **Overview of WorldPO.** WorldPO unifies two families of learning operators: policy optimization to reinforce actions (e.g., GRPO and PPO), and world modeling to model observations (e.g., SFT on environmental observations and DPO with preferences over candidate predictions). A collaborative two-fold objective couples the selected operators in a shared LLM, so the agent learns to act and predict from the same interaction experience, with world modeling providing auxiliary supervision for policy learning from environmental observations returned after actions are executed.

loss. The loss averages over all retained observation tokens, assigning each target token equal weight independent of the episode reward. Observation supervision thus extends learning from evaluating the agent's decisions to directly predicting the observations returned in response to executed actions.

Likelihood fitting uses observations already recorded in the rollouts as targets, so it requires no additional target annotation. Other objectives can address the same world-modeling task when suitable supervision is available; for example, DPO requires preferences over candidate observations predicted for the same conditioning input: the agentic history and the current generated response.

3.2 Collaborative Policy Optimization and World Modeling

WorldPO couples policy optimization and next-observation prediction within the same LLM. The policy π_θ generates responses to accomplish task goals, while the observation predictor p_θ learns to predict the next environmental observation. Sharing parameters θ allows observation supervision to train the agent used for decision-making, rather than a separate predictive model. Let $\mathcal{L}_{\text{PO}}(\theta)$ denote the selected outcome-guided policy surrogate loss introduced in Sec. 2, including its configured regularization. The collaborative two-fold loss combines both objectives over the shared parameters:

$$\mathcal{L}_{\text{WorldPO}}(\theta) = \mathcal{L}_{\text{PO}}(\theta) + \lambda \mathcal{L}_{\text{WM}}(\theta), \qquad \lambda \geq 0. \tag{3}$$

The coefficient λ controls the contribution of predictive supervision, with $\lambda = 0$ recovering the selected policy surrogate loss. The selected policy optimizer retains its advantage-estimation and credit-assignment rules. Policy advantages determine how generated responses contribute to behavioral optimization, whereas environmental observations supply world-modeling targets. Even when a response receives a negative advantage, its environmental observation remains a direct prediction target: predicting the observed consequence does not require treating that response as successful.

Each on-policy rollout batch supplies outcome feedback for $\mathcal{L}_{\text{PO}}$ and observation targets for $\mathcal{L}_{\text{WM}}$; gradients from the two losses contribute to the same parameter update. As the updated policy explores the environment, subsequent rollouts refresh the experience used for both learning objectives. This keeps the world-modeling data tied to the task states reached by the current policy, so the auxiliary objective continues to learn from experience relevant to the agent's evolving behavior. The predictive objective operates during training: at deployment, the agent receives observations from the environment and generates responses through the original interaction interface, without an additional prediction or planning stage. Appendix G details the full training procedure, causal input construction, and shared-parameter implementation, including preprocessing and environment-specific settings.

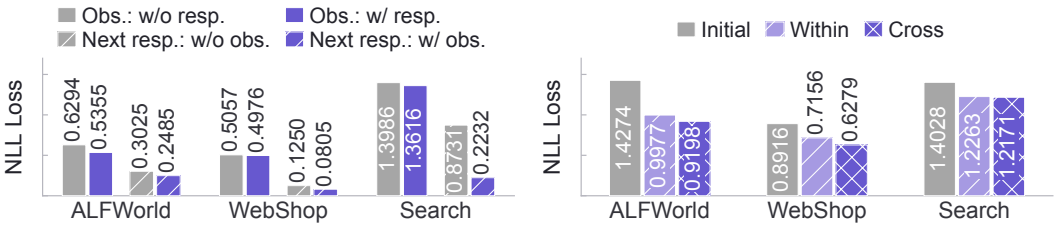

Figure 4: Held-out NLL loss (nats/token); lower values indicate better prediction. Left: Responses help predict observations, and observations help predict the next response. Right: Within-/cross-task supervision improves prediction of correctly paired observations. Study details are in Appendix C.

3.3 Theoretical Rationale for Collaborative Learning

Generated responses carry predictive information about observations. Fix the rollout policy and preprocessing. Let Q be the retained-token distribution induced by (2), as defined in Appendix D.1. For target token Y, partition its input as $S = (V, U)$: U contains the retained current generated response and V the remaining context. Let R^*_V and $R^*_{V,U}$ be the minimum expected token NLL over all conditional predictors with these inputs. Under Q and finite conditional entropies, the risks obey

$$R^*_V - R^*_{V,U} = H_Q(Y \mid V) - H_Q(Y \mid V, U) = I_Q(Y; U \mid V). \tag{4}$$

Here, H_Q and I_Q denote conditional entropy and mutual information. The risk difference compares observation prediction with and without the current generated response while preserving the remaining context. Whenever $I_Q(Y; U \mid V) > 0$, conditioning on the response reduces the minimum achievable prediction loss, identifying predictive information available in agentic experience. On held-out data, independent predictors show that current responses improve observation prediction in all three environments; the returned observations also improve prediction of the next response (Fig. 4, left).

Lower expected prediction loss reduces conditional KL error. Outcome feedback supervises policy learning on generated responses, while environmental observations condition the policy through the agentic history. World modeling additionally makes the observations prediction targets. For a normalized predictor with finite expected log loss and positive probability on the support of Q, the logarithmic score (Gneiting & Raftery, 2007) gives the conditional entropy and KL decomposition:

$$\mathbb{E}_{\mathcal{B}}[\mathcal{L}_{\mathrm{WM}}(\theta; \mathcal{B})] = H_Q(Y \mid S) + \mathbb{E}_{S \sim Q} D_{\mathrm{KL}}(Q(\cdot \mid S) \,\|\, p_\theta(\cdot \mid S)) \,. \tag{5}$$

With Q fixed, the entropy term is constant, so minimizing expected NLL minimizes conditional KL error. Unlike the policy surrogate, this objective directly scores observation tokens, including those from unsuccessful trajectories, with each retained token weighted equally across the rollout batch.

Responses and observations support complementary predictive learning. Fig. 4 (left) compares paired independent predictors trained with or without the complementary conditioning input. Each pair uses identical examples, target tokens, and optimizer budgets. Including current responses lowers ALFWorld observation NLL from 0.6294 to 0.5355; including returned observations lowers Search next-response NLL from 0.8731 to 0.2232 nats/token. Fig. 4 (right) evaluates agents trained with observation targets reassigned within or across task instances in the same environment. Both variants improve prediction of correctly paired held-out observations across all three environments; ALFWorld NLL falls from 1.4274 to 0.9977 and 0.9198, respectively. These results show complementary information in responses and observations and improved observation prediction even when exact training pairs are relaxed. Settings and held-out evaluation protocols are provided in Appendix C.

Observation gradients update the same parameters as policy gradients. For a fixed rollout batch, hold advantages, rollout likelihoods, masks, and normalizers fixed. With $g = \nabla_\theta \mathcal{L}_{\mathrm{PO}}(\theta)$ and $m = \nabla_\theta \mathcal{L}_{\mathrm{WM}}(\theta)$, differentiating (3) with respect to the shared parameters gives the joint gradient

$$\nabla_\theta \mathcal{L}_{\mathrm{WorldPO}}(\theta) = g + \lambda m. \tag{6}$$

The observation-token mask selects prediction errors whose gradients propagate through the shared LLM to form m. Thus, observation supervision and outcome feedback jointly update the policy parameters. Appendix D provides the predictive identities and local gradient analysis.

Table 1: ALFWorld success and WebShop scores and success rates (%). Purple marks the better value in each baseline–WorldPO pair at each model scale (both if tied); All denotes overall success.

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Qwen2.5-1.5B-Instruct									
Instruction-tuned base	9.38	8.33	3.85	6.67	8.33	10.53	7.81	20.22	4.69
GRPO	**68.75**	60.00	65.38	50.00	**63.64**	35.00	58.59	**73.66**	59.38
WorldPO (GRPO+WM)	67.50	**66.67**	**73.91**	**58.33**	58.33	**50.00**	**63.28**	72.58	**63.28**
RLOO	90.63	**80.00**	80.77	72.22	54.55	40.00	71.09	**77.93**	65.63
WorldPO (RLOO+WM)	**94.29**	70.00	**92.86**	**81.82**	**75.00**	**64.29**	**82.03**	74.90	**67.19**
GiGPO	90.32	**80.00**	**95.45**	63.64	58.62	**88.00**	80.47	71.25	57.03
WorldPO (GiGPO+WM)	**100.00**	**80.00**	85.71	**100.00**	**75.00**	85.71	**89.06**	**87.89**	**76.56**
Qwen2.5-3B-Instruct									
Instruction-tuned base	53.13	8.33	3.85	13.33	4.17	15.79	19.53	4.33	0.39
GRPO	90.63	**90.00**	61.54	27.78	**63.64**	**65.00**	67.19	75.44	60.16
WorldPO (GRPO+WM)	**97.14**	50.00	**78.57**	**63.64**	37.50	53.57	**69.53**	**81.65**	**69.53**
RLOO	**90.63**	**80.00**	**96.15**	72.22	**72.73**	60.00	**80.47**	70.96	57.81
WorldPO (RLOO+WM)	85.71	70.00	64.29	**90.91**	68.75	**78.57**	76.56	**81.48**	**67.97**
GiGPO	81.82	73.33	90.91	41.67	85.71	60.00	75.00	**83.79**	63.28
WorldPO (GiGPO+WM)	**100.00**	**80.00**	**100.00**	**90.91**	**87.50**	**82.14**	**92.19**	83.75	**71.09**

4 Experiments

4.1 Experimental Setting

We evaluate Qwen2.5-1.5B/3B-Instruct on **ALFWorld** household tasks (Shridhar et al., 2021) and **WebShop** product selection (Yao et al., 2022), and Qwen2.5-3B/7B-Instruct on **search-based question answering** (Jin et al., 2025). Observations describe object states and action outcomes in ALFWorld, present product attributes and webpage content in WebShop, and provide retrieved passages in Search. Search models train on NQ and HotpotQA and are also evaluated on TriviaQA, PopQA, 2WikiMultiHopQA, and MuSiQue to evaluate generalization beyond the training datasets.

On ALFWorld and WebShop, we compare GRPO (Shao et al., 2024), RLOO (Ahmadian et al., 2024), and GiGPO (Feng et al., 2025) with their WorldPO variants and include instruction-tuned baselines. World modeling uses supervised fine-tuning (SFT) on returned observations; Search uses GRPO + SFT. Search baselines include GiGPO, R1-Instruct, Search-R1 (GRPO) (Jin et al., 2025), ZeroSearch (Sun et al., 2025), and StepSearch (Zheng et al., 2025). We report ALFWorld success, WebShop score and success, and search accuracy per dataset and as a six-dataset equal-weight average (Avg.). Evaluation protocols and detailed training configurations are given in Appendices B and G.

4.2 Main Results

WorldPO improves task completion across diverse observation types. Despite these differences, adding world modeling improves task completion in all three environments. With 1.5B GRPO, success rises from 58.59% to 63.28% on ALFWorld and from 59.38% to 63.28% on WebShop (Tab. 1). The benefit also appears with RLOO, raising 1.5B ALFWorld success from 71.09% to 82.03%. On Search, 3B WorldPO achieves 40.71% average accuracy, versus 35.30% for Search-R1 (GRPO) (Fig. 5). These results establish observation supervision as a useful addition across tasks with different forms of environmental information under the same observation-prediction objective.

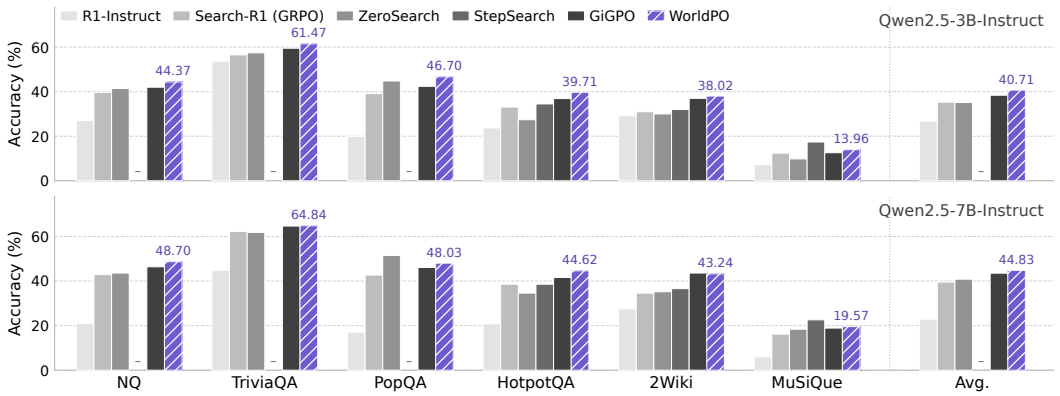

Figure 5: Search task performance of Qwen2.5-3B-Instruct (top) and Qwen2.5-7B-Instruct (bottom). Gray bars show published baselines; purple hatched bars show WorldPO. Baseline results are taken directly from the Search experiments in Search-R1 (Jin et al., 2025) and GiGPO (Feng et al., 2025).

World modeling complements action-level policy optimization. GiGPO refines policy optimization through action-level credit assignment, while world modeling adds direct supervision on returned observations. Combining them raises ALFWorld success from 80.47% to 89.06% at 1.5B and from 75.00% to 92.19% at 3B. At 3B, WebShop success also increases from 63.28% to 71.09% (Tab. 1). On ALFWorld, the 1.5B combined policy exceeds the 3B GiGPO baseline. These improvements are consistent with the collaborative-learning rationale in Sec. 3.3: action-level credit assignment refines policy supervision on generated responses, while world modeling supplies complementary supervision from observations to the same model through the auxiliary observation-prediction loss.

Search improvements extend beyond the training datasets. Although Search training uses only NQ and HotpotQA, 3B WorldPO outperforms GiGPO on all four additional evaluation datasets: TriviaQA, PopQA, 2WikiMultiHopQA, and MuSiQue. For example, accuracy rises from 42.40% to 46.70% on PopQA and from 12.60% to 13.96% on MuSiQue. These improvements cover both single-hop and multi-hop questions, supporting generalization beyond the training datasets. At 7B, WorldPO also exceeds Search-R1 (GRPO) on all four additional datasets and reaches 44.83% average accuracy, compared with 39.55% for Search-R1 and 43.55% for GiGPO (Fig. 5). Thus, the benefits extend beyond the training datasets and persist across model scales for single- and multi-hop QA.

5 Understanding Holistic Agentic Optimization

We investigate why world modeling improves agent capabilities and how these improvements are reflected in agent behavior. We first examine the value of task-relevant environmental knowledge through observation-target reassignment and weighting, then analyze changes in action reliability, completion efficiency, and task-directed reasoning by comparison with policy-only GRPO training.

5.1 Benefits from Task-Relevant Knowledge

We investigate whether WorldPO benefits from learning action–observation correspondences or acquiring environmental knowledge. Target reassignment within and across tasks tests the roles of exact pairing and task relevance. We also compare uniform, correctness-weighted, and similarity-weighted supervision to assess the value of answer-relevant observations and successful experience.

World modeling benefits from task-relevant environmental knowledge. We compare three world-modeling variants alongside GRPO across ALFWorld, WebShop, and Search: *WorldPO* uses the observation following each action; *within-task* draws an observation from another transition in the same task instance; and *cross-task* draws from a different task instance in the same environment. In Fig. 6, within-task supervision achieves 79.69% ALFWorld success, versus 72.66% for both WorldPO and cross-task supervision, with absolute improvements of 4.69% and 1.58% over cross-

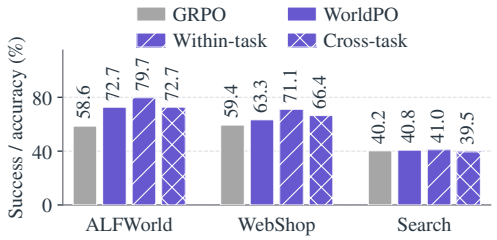

Figure 6: Comparison of world modeling under supervision from correctly paired, random within-task, and random cross-task observation targets.

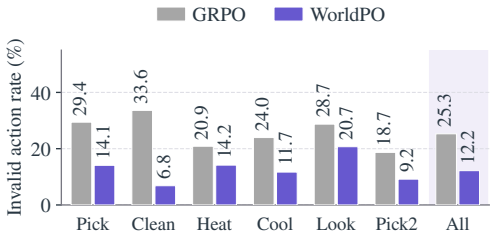

Figure 7: **Invalid action rate comparison.** Frequency of invalid actions ("Nothing happens") for GRPO-only and WorldPO agents in ALFWorld.

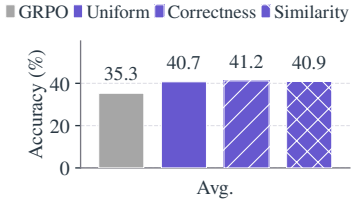

Figure 8: Search accuracy under different observation-weighting rules for the world-modeling loss.



Figure 9: **Agentic reasoning efficiency.** Average number of interaction turns for GRPO and WorldPO across three environments, using all episodes (left) and jointly solved tasks (right).

task supervision on WebShop and Search. These results highlight the value of task-relevant knowledge beyond exact action–observation pairing. Full protocols and results appear in Appendices B and F.

Observations from successful experience provide more useful supervision. We examine which observations best support agent learning by comparing uniform supervision ($w_{t+1} = 1$) with two weighting strategies on Search: *correctness-weighted* supervision uses the trajectory return, $w_{t+1} = r(\tau)$; *similarity-weighted* supervision uses $w_{t+1} = \mathrm{sim}(o_{t+1}, y^\star)$, the cosine-based semantic relevance to the reference answer. The world-modeling variants use Qwen2.5-3B-Instruct with 200 updates and $\lambda = 0.1$. Fig. 8 shows that similarity weighting improves accuracy from 40.7% with uniform supervision to 40.9%, while correctness weighting reaches 41.2%. These gains suggest that answer-relevant observations help, but successful trajectories provide richer task-solving knowledge than answer similarity alone captures. Appendix B.4 gives detailed settings and per-dataset results.

5.2 More Reliable and Efficient Agent Behavior

Next, we examine how world modeling shapes agent behavior by comparing WorldPO (GRPO + WM) with GRPO-only training. We analyze invalid actions, the number of turns required to complete tasks, and a case study of task-directed tool use. Together, these analyses examine how agents apply environmental knowledge during task execution. Full analysis details are provided in Appendix I.

WorldPO takes fewer invalid actions. We measure the invalid action rate in ALFWorld as the fraction of action attempts for which the environment returns "Nothing happens." As shown in Fig. 7, this rate falls from 25.28% with GRPO to 12.19% with WorldPO. This reduction in environment-rejected action attempts suggests that agents trained with world modeling better understand the environment's action requirements and tend to choose actions consistent with those requirements.

WorldPO completes jointly solved tasks in fewer turns. We compare the average number of interaction turns across all episodes and across tasks solved by both methods. Fig. 9 shows that WorldPO uses fewer turns on jointly solved tasks in all three environments. This pattern suggests that agents trained with world modeling tend to follow more direct routes to task completion, consistent with more effective use of environmental knowledge on tasks that both methods successfully complete.

Case study: WorldPO connects observed progress to task-directed tool use. Fig. 10 compares two successful trajectories for the task "examine the bowl with the desklamp." After the same initial

ALFWorld: Task and shared action–observation history

Task. "examine the bowl with the desklamp."
Shared history, $t = 0$–3 (observations summarized; zero-based turns).

$t = 0$	`go to desk 1`	Bowl and desk lamp are on the desk.
$t = 1$	`take bowl 1 from desk 1`	Bowl is picked up.
$t = 2, 3$	`examine bowl 1`	Normal, empty bowl; same observation twice.

GRPO

Repeated inspections.
$t = 4$: `examine bowl 1`
$t = 5$: `examine desk 1`
$t = 6$: `examine desklamp 1`
"Nothing happens."
$t = 7$–8: `examine bowl 1` (twice)
Reasoning excerpt at $t = 4$: "The user has already examined the bowl and found nothing inside."
$t = 9$: `use desklamp 1`
Success in 10 turns.
Five extra inspections before using the task-required lamp.

WorldPO

Connect observed state to tool use.
Reasoning excerpt at $t = 4$: "I have already taken steps to pick up the bowl from the desk and examine it. Now, I need to find a way to **use the desklamp to examine the bowl further.**"
$t = 4$: **`use desklamp 1`**
"You turn on the desklamp 1."
Success in 5 turns.
Connects the acquired object and observed progress to the lamp's task-relevant function.

Figure 10: **ALFWorld case study of task-directed tool use.** GRPO and WorldPO reasoning and actions after a common history, with summarized observations and verbatim excerpts of reasoning.

actions and observations, both agents have picked up the bowl and examined it twice. WorldPO then reasons that it needs to use the lamp and executes `use desklamp 1`, completing the task. GRPO instead performs five additional inspections before using the same tool. This contrast suggests that world modeling encourages agents to connect observed progress with task requirements and select goal-directed actions. Further cases and the full ALFWorld interaction trace appear in Appendix I.

6 Related Work

Policy optimization improves agent behavior through task feedback, with GRPO and GiGPO differing in advantage estimation and credit assignment (Shao et al., 2024; Feng et al., 2025). World modeling provides complementary supervision from environment observations. Prior work uses this supervision for experience-based initialization (Zhang et al., 2025) or combines it directly with policy learning: DyMo couples RLOO with next-state supervision, while ECHO augments GRPO with environment-token supervision (Guo et al., 2025; Shrivastava et al., 2026). WorldPO builds on these shared-model approaches to examine how task-relevant observation supervision complements policy optimization across diverse agentic environments. The full discussion of related work is provided in Appendix A.

7 Conclusion

We presented WorldPO, a framework for holistic agentic learning that combines policy optimization and world modeling in the same LLM using the same on-policy trajectories. Experiments in three agentic environments with different types of observations demonstrate improved task performance across model scales and policy optimizers. In particular, world modeling complements GiGPO's action-level credit assignment and improves Search performance on single-hop and multi-hop datasets not used for training. Our analyses suggest that world modeling supplies task-relevant environmental knowledge beyond exact action–observation correspondence, with successful experience providing particularly useful supervision. Behavioral analyses further show lower invalid action rates and fewer turns on jointly solved tasks compared with policy-only GRPO training. Future work will explore adaptive observation selection and extend collaborative learning to richer multimodal environments.

AI USE STATEMENT

We used generative AI tools to polish the manuscript, including improving grammar, phrasing, clarity, and consistency of presentation. We reviewed and revised the AI-assisted text for accuracy and consistency with the methods, results, and intended claims. The authors take responsibility for the final content of this paper, including the text, analyses, and conclusions.

ETHICS STATEMENT

WorldPO aims to improve agentic learning through collaborative policy optimization and world modeling. Our experiments use publicly available ALFWorld, WebShop, and question-answering benchmarks, and we do not identify ethical concerns specific to these benchmark experiments. More capable autonomous agents may nevertheless be misused, and storing agentic histories and environment observations can raise privacy and data-governance concerns in applications with sensitive information. We encourage sandboxed execution, restricted tool access, privacy-preserving logging, and transparent reporting of deployment scope and limitations.

REPRODUCIBILITY STATEMENT

Sec. 3 describes the learning objectives, and Sec. 4 specifies the experimental setting. Appendix G provides the training algorithm, input and target construction, and training and evaluation configurations (Tab. 28), including hyperparameters, decoding settings, and parallelism. Appendix B specifies the metrics, evaluation protocols, and supervision controls; Appendix D gives the theoretical assumptions and proofs. Appendices C, E, and F document the predictive and gradient diagnostics and shared-task behavioral comparisons.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *ACL*, 2024.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *NeurIPS*, 2024.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Nghi DQ Bui. Building effective ai coding agents for the terminal: Scaffolding, harness, context engineering, and lessons learned. *arXiv preprint arXiv:2603.05344*, 2026.

Shiqi Chen, Tongyao Zhu, Zian Wang, Jinghan Zhang, Kangrui Wang, Ruochen Zhou, Siyang Gao, Teng Xiao, Yee Whye Teh, Junxian He, and Manling Li. Why Do LLM Agents Fail in Exploring New Environments? A World-Modeling Perspective. *arXiv preprint arXiv:2510.15047*, 2026. doi: 10.48550/arXiv.2510.15047. URL https://arxiv.org/abs/2510.15047v2.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of machine learning research*, 24(240):1–113, 2023.

Yunshu Du, Wojciech M Czarnecki, Siddhant M Jayakumar, Mehrdad Farajtabar, Razvan Pascanu, and Balaji Lakshminarayanan. Adapting auxiliary losses using gradient similarity. *arXiv preprint arXiv:1812.02224*, 2018.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. In *NeurIPS*, 2025.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Shangmin Guo, Omar Darwiche Domingues, Raphaël Avalos, Aaron Courville, and Florian Strub. World modelling improves language model agents. *arXiv preprint arXiv:2506.02918*, 2025.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

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

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models, 2019. URL https://arxiv.org/abs/1911.12247. ICLR 2020.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *ICLR Deep RL Meets Structured Prediction Workshop*, 2019.

Yixia Li, Hongru Wang, Jiahao Qiu, Zhenfei Yin, Dongdong Zhang, Cheng Qian, Zeping Li, Xiaoteng Ma, Guanhua Chen, and Heng Ji. From word to world: Can large language models be implicit text-based world models? In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8084–8111, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. In *NeurIPS*, 2024.

Yujia Qin, Yining Ye, Junjie Fang, Haoming Wang, Shihao Liang, Shizuo Tian, Junda Zhang, Jiahao Li, Yunxin Li, Shijue Huang, et al. Ui-tars: Pioneering automated gui interaction with native agents. *arXiv preprint arXiv:2501.12326*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In *ICLR*, 2021.

Vaishnavi Shrivastava, Piero Kauffmann, Ahmed Awadallah, and Dimitris Papailiopoulos. Echo: Terminal agents learn world models for free. *arXiv preprint arXiv:2605.24517*, 2026.

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

Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Yiping Lu, Zhengyuan Yang, Lijuan Wang, et al. Vagen: Reinforcing world model reasoning for multi-turn vlm agents. In *NeurIPS*, 2025a.

Siyin Wang, Zhaoye Fei, Qinyuan Cheng, Shiduo Zhang, Panpan Cai, Jinlan Fu, and Xipeng Qiu. World modeling makes a better planner: Dual preference optimization for embodied task planning. In *ACL*, pp. 21518–21537, 2025b.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *NeurIPS*, 2024.

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

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In *NeurIPS*, 2022.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *NeurIPS*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In *NeurIPS*, volume 33, pp. 5824–5836, 2020.

Xiao Yu, Baolin Peng, Ruize Xu, Yelong Shen, Pengcheng He, Suman Nath, Nikhil Singh, Jiangfeng Gao, and Zhou Yu. Reinforcement world model learning for llm-based agents. *arXiv preprint arXiv:2602.05842*, 2026.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, et al. Agent learning via early experience. *arXiv preprint arXiv:2510.08558*, 2025.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization. In *EMNLP*, pp. 21805–21830, 2025.

Fangqi Zhu, Zhengyang Yan, Zicong Hong, Quanxin Shou, Xiao Ma, and Song Guo. Wmpo: World model-based policy optimization for vision-language-action models. In *ICLR*, volume 2026, pp. 62486–62502, 2026.

Yuxin Zuo, Zikai Xiao, Li Sheng, Fei Huang, Jianhong Tu, Yuxuan Liu, Tianyi Tang, Xiaomeng Hu, Yang Su, Qingfeng Lan, et al. Qwen-agentworld: Language world models for general agents. *arXiv preprint arXiv:2606.24597*, 2026.

APPENDIX

A Related Work . . . 14

B Evaluation Protocol and Supervision Controls . . . 15
- B.1 Comparison Groups and Metrics . . . 15
- B.2 Complete Search Results . . . 15
- B.3 Full-Suite Search Evaluation . . . 15
- B.4 Observation-Weighting Strategies . . . 16
- B.5 Observation-Target Reassignment Controls . . . 17

C Predictive Diagnostics . . . 18
- C.1 Independent Predictors . . . 18
- C.2 Shared-Agent Observation Prediction . . . 19

D Theoretical Details of Collaborative Learning . . . 19
- D.1 The Retained-Token Predictive Objective . . . 19
- D.2 Compatible Collaborative Updates . . . 21
- D.3 Additional Progress Relative to Policy-Only Learning . . . 22
- D.4 Optimizer Scope . . . 22

E Gradient Diagnostics . . . 23
- E.1 Observation Targets and Model Depth Change Gradient Strength . . . 23
- E.2 Definitions and Training-Stage Measurements . . . 23
- E.3 Reward-Source and Depth Diagnostics . . . 24

F Behavior on Shared Evaluation Tasks . . . 27
- F.1 Evaluation and Paired Statistics . . . 27
- F.2 ALFWorld and WebShop Checkpoint Comparisons . . . 27
- F.3 Search Dataset Outcomes and Paired Comparisons . . . 29
- F.4 New and Lost Successes on Paired Episodes . . . 29

G WorldPO Training Procedure and Implementation . . . 32
- G.1 Training and Evaluation Configurations . . . 32
- G.2 Training Procedure . . . 33
- G.3 Causal Inputs and Prediction Targets . . . 33
- G.4 Shared-Parameter Updates . . . 33

H Extended Preliminaries . . . 34
- H.1 Interaction and Trajectory Distribution . . . 34
- H.2 Policy Objective and Gradient . . . 34
- H.3 World-Modeling Objectives . . . 34

I Case Studies of Agentic Reasoning . . . 35
- I.1 ALFWorld: Task Progress and Repeated Examination . . . 35
- I.2 WebShop: Option Selection and Purchase Completion . . . 36
- I.3 Search: Evidence Acquisition and Answer Completion . . . 36
- I.4 ALFWorld Reasoning Excerpts and Complete Action Sequences . . . 37

A Related Work

We review how feedback drives policy updates, how predictive objectives learn environment dynamics, and how policy learning incorporates world models.

Policy optimization for language agents. Policy-learning methods differ in how they turn feedback into parameter updates. PPO uses a clipped surrogate objective (Schulman et al., 2017); GRPO estimates group-relative advantages without a learned critic (Shao et al., 2024), while RLOO constructs a leave-one-out baseline from other sampled responses (Ahmadian et al., 2024; Kool et al., 2019). DAPO refines clipping and sampling (Yu et al., 2025), and Dr. GRPO addresses normalization-induced biases (Liu et al., 2025). Preference objectives such as DPO and SimPO instead learn from response comparisons (Rafailov et al., 2023; Meng et al., 2024). Interactive tasks additionally require feedback to guide decisions across turns: Search-R1 trains reasoning interleaved with retrieval (Jin et al., 2025), and GiGPO combines trajectory-level comparisons with action comparisons at repeated environment states (Feng et al., 2025). These developments concern update rules, advantage estimation, and credit assignment. WorldPO retains the selected policy surrogate loss and adds a complementary source of supervision: predicting the environmental observations produced by the agent's behavior.

Learning objectives for world modeling. World-model learning can supervise observations, latent dynamics, or judgments of predictive correctness. For textual observations, teacher-forced maximum likelihood yields a token-level cross-entropy objective for supervised fine-tuning, conditioned on the agentic history and current action (Li et al., 2026). Variational objectives combine observation reconstruction with latent-transition regularization (Hafner et al., 2019). Prediction can instead operate in representation space: SPR matches predicted future features to target-encoder representations (Schwarzer et al., 2020), while V-JEPA 2 predicts representations of masked video patches and subsequently trains an action-conditioned latent predictor (Assran et al., 2025). Contrastive transition losses distinguish observed successor representations from negative samples without pixel reconstruction (Kipf et al., 2019); diffusion denoising learns conditional distributions over future observations (Alonso et al., 2024). Prediction quality can also define a preference or reward rather than a token-exact target. D^2PO optimizes preferences over next-state descriptions (Wang et al., 2025b), whereas RWML applies GRPO to rewards based on semantic agreement between predicted and realized states (Yu et al., 2026). These objectives differ in the information they require the predictor to preserve. WorldPO selects observation-token likelihood for collaborative training on the policy's own interactions.

World modeling in policy learning. Predictive objectives support policy learning through initialization, simulated experience, or auxiliary supervision. Early Experience and SPA learn from explored states before subsequent policy RL (Zhang et al., 2025; Chen et al., 2026); RWML similarly precedes task-success training with a world-model RL stage (Yu et al., 2026). WMPO generates imagined experience with a visual world model for policy optimization (Zhu et al., 2026), and Qwen-AgentWorld uses a separate language world model as an agentic RL environment (Zuo et al., 2026). VAGEN instead rewards explicit state estimation and transition prediction during multi-turn visual-agent RL (Wang et al., 2025a). The closest objective-level precedents directly update a shared agent: DyMo combines online RLOO with next-state cross-entropy for tool use (Guo et al., 2025), and ECHO adds environment-token cross-entropy to GRPO on the same multi-turn terminal rollouts (Shrivastava et al., 2026). Building on these routes from prediction to policy learning, WorldPO formulates collaborative optimization over generated responses and environmental observations. We examine observation-target pairing, predictive conditioning, and gradient geometry across textual interaction environments, relating the content of observation supervision to its effect on policy learning.

Table 2: **Full-training endpoint success by observation target.** Within-task and cross-task targets follow the original control protocol in each environment. Absolute differences are computed before rounding where counts are available. Search uses the reported equal-weight dataset averages. These endpoints are separate from the common-task reevaluation.

Environment	Within-task success (%)	Cross-task success (%)	Difference (%)
ALFWorld	**79.69**	72.66	+7.03
WebShop	**71.09**	66.41	+4.69
Search	**39.52**	38.17	+1.35

B Evaluation Protocol and Supervision Controls

B.1 Comparison Groups and Metrics

We use environment seed 0 and auxiliary weight $\lambda = 0.1$. Tab. 1 covers Qwen2.5-1.5B/3B-Instruct on ALFWorld and WebShop; Tab. 3 covers Qwen2.5-3B/7B-Instruct on search tasks. The understanding controls use the 1.5B model for ALFWorld and WebShop and the 3B model for Search. Their final evaluations are at 150, 150, and 200 updates, respectively. ALFWorld and WebShop evaluate 128 episodes; the Search table evaluates 51,588 questions from six datasets.

The ALFWorld correctly paired, within-task, and cross-task observation-target reassignment study has its own newly trained correctly paired control. Its checkpoints differ from the WorldPO runs in the main tables. Tab. 2 reports the within-task and cross-task training endpoints; Appendix F reports their common-task reevaluations. Execution settings are specified below.

Fig. 6 combines these control endpoints with the main-table GRPO results for ALFWorld and WebShop and the main-table WorldPO result for WebShop. Its ALFWorld WorldPO bar uses the dedicated target-binding control; its Search GRPO and WorldPO bars use the per-question evaluations in Tab. 4. The Search bars average six dataset rates equally, excluding Bamboogle, as does Avg-6 in that table.

For Search, per-dataset success is the fraction of questions whose native episode score is at least 1. The Search table's Avg is the equal-weight arithmetic mean of NQ, TriviaQA, PopQA, HotpotQA, 2WikiMultiHopQA, and MuSiQue, computed before rounding. Our 3B WorldPO entries use the logged per-dataset rates, while our 7B entries use per-question records. Published search baselines retain their reported dataset rates. The within-task and cross-task controls below retain their separate seven-dataset protocol. Framework-wide batch aggregates remain separate from this equal-weight average. WebShop success is the fraction of episodes achieving a shopping score of 1; the mean fractional shopping score is reported separately.

B.2 Complete Search Results

Tab. 3 gives the complete 3B and 7B search comparisons, including all values plotted in Fig. 5. Unavailable results remain unreported; in particular, StepSearch has no six-dataset average because its single-hop results are unavailable.

B.3 Full-Suite Search Evaluation

We reevaluate the Qwen2.5-3B-Instruct GRPO and WorldPO (GRPO+SFT) checkpoints after 200 updates, reporting 51,588 questions from six datasets: 3,610 NQ, 11,313 TriviaQA, 14,267 PopQA, 7,405 HotpotQA, 12,576 2WikiMultiHopQA, and 2,417 MuSiQue examples. Both methods use greedy decoding, retrieval top-$k = 3$, a four-turn horizon, prompt/response limits of 4096/512 tokens, validation batches of 512, and one evaluation trajectory per question.

Tab. 4 reports success rates computed from individual question outcomes. Avg-6 is the equal-weight mean of the six full-precision dataset rates. These checkpoint reevaluations supply the Search GRPO and WorldPO bars in Fig. 6, using the same six-dataset average. Tab. 3 reports the historical training evaluations.

Table 3: Performance on search tasks (%). Avg. is the equal-weight mean of all six dataset accuracies; NQ and HotpotQA are the training-domain datasets. Search-R1 (GRPO) uses the Instruct results in Table 3 of Jin et al. (2025); other published baselines are reported as collected by Feng et al. (2025). WorldPO combines GRPO with SFT on returned observations. Dark purple marks the highest reported value in each column within the same model scale; – indicates unavailable results.

Method	**Single-hop QA**			**Multi-hop QA**			**Avg.**
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	
Qwen2.5-3B-Instruct							
R1-Instruct	27.00	53.70	19.90	23.70	29.20	7.20	26.78
Search-R1 (GRPO)	39.70	56.50	39.10	33.10	31.00	12.40	35.30
ZeroSearch	41.40	57.40	44.80	27.40	30.00	9.80	35.13
StepSearch	–	–	–	34.50	32.00	**17.40**	–
GiGPO	42.00	59.50	42.40	36.90	37.00	12.60	38.40
WorldPO (GRPO + SFT)	**44.37**	**61.47**	**46.70**	**39.71**	**38.02**	13.96	**40.71**
Qwen2.5-7B-Instruct							
R1-Instruct	21.00	44.90	17.10	20.80	27.50	6.00	22.88
Search-R1 (GRPO)	42.90	62.30	42.70	38.60	34.60	16.20	39.55
ZeroSearch	43.60	61.80	**51.50**	34.60	35.20	18.40	40.85
StepSearch	–	–	–	38.60	36.60	**22.60**	–
GiGPO	46.40	64.70	46.10	41.60	**43.60**	18.90	43.55
WorldPO (GRPO + SFT)	**48.70**	**64.84**	48.03	**44.62**	43.24	19.57	**44.83**

Table 4: **Full-suite Search checkpoint evaluation.** Per-question accuracy (%) for GRPO and WorldPO (GRPO+SFT). Avg-6 equally weights the six datasets.

Dataset	GRPO	WorldPO (GRPO+SFT)
NQ	44.16	45.76
TriviaQA	60.33	61.28
PopQA	44.20	45.93
HotpotQA	39.61	39.92
2WikiMultiHopQA	38.12	38.12
MuSiQue	15.06	13.49
Avg-6	40.25	40.75

B.4 Observation-Weighting Strategies

The comparison in Fig. 8 varies observation weights within likelihood-based world modeling. Uniform weighting assigns every observation weight one; correctness-weighted supervision uses the trajectory return $r(\tau)$; similarity-weighted supervision uses cosine-based semantic relevance between an observation and the reference answer. Similarity weights are computed from reference answers during training. The three world-modeling variants use Qwen2.5-3B-Instruct with 200 updates and $\lambda = 0.1$. Uniform uses the current WorldPO-3B result in Tab. 3; the two weighted variants use the historical weighting runs. The GRPO baseline is the 3B-Instruct GRPO result reported by Search-R1 (Jin et al., 2025). This comparison uses the six QA benchmarks in Tab. 5. Avg-6 averages the six dataset rates before display rounding, using the native logged rates for Uniform and the published or historical one-decimal percentages for the other methods.

All three strategies exceed the GRPO baseline on each dataset. Correctness-weighted and similarity-weighted supervision yield absolute accuracy improvements of 0.44% and 0.16% over Uniform, respectively. Correctness weighting achieves the highest average, supporting the value of observations associated with successful task execution.

Table 5: **Observation-weighting results on six QA benchmarks.** Accuracy (%) with Qwen2.5-3B-Instruct. World-modeling variants use 200 updates and $\lambda = 0.1$; GRPO is from Search-R1. Avg-6 is computed before display rounding.

Weighting	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Avg-6
GRPO	39.7	56.5	39.1	33.1	31.0	12.4	35.3
Uniform	44.4	61.5	46.7	39.7	38.0	14.0	40.7
Correctness-weighted	45.8	61.3	46.8	39.5	38.5	15.0	41.2
Similarity-weighted	45.0	60.7	45.4	39.5	39.2	15.4	40.9

B.5 Observation-Target Reassignment Controls

Correctly paired supervision uses correctly paired observation targets. Within-task supervision assigns an observation from another transition in the same task; cross-task supervision assigns an observation from a different task identity. The task identity rule permits a donor from the same semantic task category. The policy still conditions on the original environment observations, and its generated responses, executable actions, and rewards are unchanged. When no legal donor is available, the builder retains the original target. Tab. 6 reports how often within-task supervision retains the original target. The completed cross-task runs have zero fallback; ALFWorld cross-task supervision reassigns 732,160 targets and WebShop cross-task supervision reassigns 172,800.

Table 6: **Within-task reassignment retains the original observation target when no legal donor is available.** Counts accumulate over the completed within-task training run in each environment.

Environment	Original-target fallbacks	Total targets
ALFWorld	1,946	710,208
WebShop	37,916	165,792
Search	117,161	393,068

Tab. 7 reports all Search within-task and cross-task dataset outcomes. Within-task supervision uses the legacy implementation for updates 1–50 and verl 0.9 thereafter; cross-task supervision uses verl 0.9 with FSDP2 throughout. Within-task and cross-task supervision use two-way rollout tensor parallelism. The original correctly paired run uses one-way tensor parallelism and logs dataset-level batch means. These protocol identities distinguish the new per-question results from the historical main-table rates.

Table 7: **Search outcomes with within-task and cross-task observation targets.** The within-task and cross-task controls use the same seven evaluation datasets. Avg averages the seven success rates equally.

		Within-task target		Cross-task target	
Dataset	Questions	Successful	Success (%)	Successful	Success (%)
NQ	3,610	1,649	45.68	1,646	45.60
TriviaQA	11,313	7,011	61.97	6,916	61.13
PopQA	14,267	6,745	47.28	6,672	46.77
HotpotQA	7,405	2,883	38.93	2,821	38.10
2WikiMultiHopQA	12,576	4,724	37.56	4,053	32.23
MuSiQue	2,417	358	14.81	313	12.95
Bamboogle	125	38	30.40	38	30.40
Avg			39.52		38.17

WebShop validates before training and every five updates. Tab. 8 gives success counts and shopping scores. The cross-task evaluation samples goals from a persistent seeded generator, so successive checkpoints can evaluate different episodes.

Table 8: **WebShop control endpoints.** Success and shopping task score are distinct metrics, reported as percentages.

Auxiliary target	Successful / evaluated	Success (%)	Task score (%)
Within-task observation	91 / 128	71.09	82.58
Cross-task observation	85 / 128	66.41	78.53

C Predictive Diagnostics

C.1 Independent Predictors

We train independent predictors on frozen, task-disjoint transition splits. ALFWorld and Search initialize from their GRPO update-150 and update-200 checkpoints; WebShop initializes from its WorldPO checkpoint trained with correctly paired observation targets at update 150. Each arm fine-tunes the full model for one epoch with learning rate 10^{-6}, weight decay 0.01, micro-batch size 1, gradient accumulation 8, maximum gradient norm 1, bfloat16, and target-token mean cross-entropy. Within each pair, splits, shuffling, target masks, token counts, and optimizer budgets are identical.

Observation-prediction improvement is the initial-minus-final held-out observation NLL of the predictor conditioned on the agentic history and current generated response. Response-conditioned observation gain is the final observation NLL of the history-only predictor minus that of the history-and-generated-response-conditioned predictor. Observation-conditioned response gain compares prediction of the recorded next generated response: it is the final NLL without the returned observation minus that with it. Values are in nats per target token. These are predictive comparisons on recorded behavior; their different initializations and training budgets are retained when interpreting cross-environment magnitudes.

Table 9: **Per-arm diagnostic accounting.** The paired predictors use the same examples, updates, and target-token budget.

Environment	Study	Train / held-out examples	Updates	Train / held-out tokens
ALFWorld	Observation / response-conditioned gains	4,731 / 970	592	124,476 / 23,600
ALFWorld	Observation-conditioned response gain	4,523 / 922	566	29,741 / 5,956
Search	Observation / response-conditioned gains	711 / 149	89	340,487 / 70,683
Search	Observation-conditioned response gain	711 / 149	89	82,082 / 17,868
WebShop	Observation / response-conditioned gains	1,889 / 439	237	514,685 / 123,749
WebShop	Observation-conditioned response gain	1,683 / 389	211	10,231 / 2,467

Observation-conditioned response gain excludes transitions without a following generated response. Train/held-out task counts are 74/17 for ALFWorld, 415/97 for Search, and 101/25 for WebShop. There are twelve fitted predictors across the six paired studies.

Responses and observations provide complementary predictive information. We separately train diagnostic predictors on task-disjoint transition splits, changing one conditioning input at a time. Adding the current generated response to the agentic history reduces held-out observation NLL by 0.0939, 0.0081, and 0.0370 nats per target token on ALFWorld, WebShop, and Search, respectively. In the reverse direction, adding the returned observation reduces the NLL of the recorded next generated response by 0.0540, 0.0445, and 0.6499 (Fig. 11). Each comparison uses separately trained predictors with the same examples, target tokens, and optimizer budget. The positive differences identify information available in each part of an interaction: a generated response helps predict what the environment returns, and the returned observation helps predict the next response. These paired prediction tasks identify useful conditioning structure in recorded interactions and support predicting observations from both history and the current generated response.

Table 10: **Conditioning information improves prediction in the diagnostic studies.** Values are held-out NLL reductions in nats per target token from separately trained diagnostic predictors. The first two columns concern observation prediction; the last concerns prediction of the recorded next generated response.

Environment	Observation NLL reduction	Response-conditioned observation gain	Observation-conditioned response gain
ALFWorld	**0.8951**	**0.0939**	0.0540
WebShop	0.1111	0.0081	0.0445
Search	0.1311	0.0370	**0.6499**

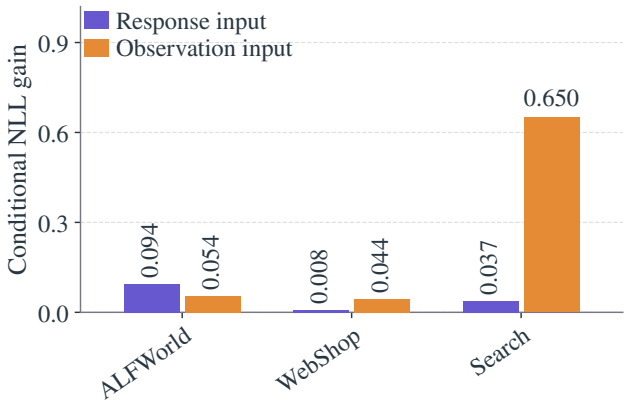

Figure 11: Conditional information in generated responses and returned observations. Bars show held-out NLL reductions from adding the current generated response to observation prediction (purple), or the returned observation to prediction of the recorded next generated response (orange). Each pair uses independently fitted predictors with identical examples, target tokens, and optimizer budgets. Values are in nats per target token.

C.2 Shared-Agent Observation Prediction

The original ALFWorld likelihood study scores the shared agent on 970 held-out examples from 17 tasks, totaling 23,600 target tokens. Tab. 12 shows the available within-task and cross-task checkpoint sequences under all three observation-target pairings. These losses are evaluated on one frozen pool rather than on each stage's training batch. Cross-task supervision continues to assign lower NLL to original observations than to within-task and cross-task donors even while training against cross-task targets.

The extended study evaluates eight available checkpoints using a common feature construction within each environment (Tab. 13). It scores the initial model, within-task, and cross-task checkpoints for Search and WebShop, and the initial model and cross-task checkpoints for ALFWorld, with full-precision total token loss divided by total retained target tokens. This extended study uses bfloat16 scoring; Tab. 11 retains the original ALFWorld scoring results. Each environment's checkpoint comparisons share the recorded feature and target-token hashes.

D Theoretical Details of Collaborative Learning

D.1 The Retained-Token Predictive Objective

Fix a behavior policy and the preprocessing rules in Appendix G. Sample an auxiliary batch $\mathcal{B}$ conditional on its containing at least one valid target token, and let $N_{\mathcal{B}} = \sum_{(h_t, y_t, o_{t+1}) \in \mathcal{B}} |o_{t+1}|$. Select a token position uniformly from the $N_{\mathcal{B}}$ retained observation-token positions, counting repeated transitions separately. For a selected position j in observation o_{t+1}, set $Y = o_{t+1,j}$ and let S be

Table 11: The ALFWorld target-binding comparison uses one training protocol. Prediction scores use one held-out pool; NLL is in nats per target token.

Training target	Success (%)	True-observation NLL	Within-task donor NLL
Correctly paired	72.66	1.0033	1.2879
Within-task	**79.69**	0.9977	1.0316
Cross-task	72.66	**0.9198**	**1.0000**

Table 12: **Training on reassigned observation targets reduces original-observation NLL.** The original ALFWorld scoring study uses identical held-out examples across these checkpoints. NLL is in nats per target token.

Training arm	Checkpoint	Original target	Within-task target	Cross-task target
Initial model	0	1.4274	1.3987	2.3057
Within-task target	50	1.0914	1.0980	1.7977
Within-task target	100	0.9934	1.0262	1.6761
Within-task target	150	0.9977	1.0316	1.6894
Cross-task target	50	1.0406	1.1132	1.6193
Cross-task target	100	0.9672	1.0295	1.4755
Cross-task target	150	0.9198	1.0000	1.4196

the exact input used to score it, including the retained history h_t, generated response y_t, preceding observation tokens $o_{t+1,<j}$, and positional information. This construction defines $Q(S, Y)$ and gives

$$\begin{aligned}\mathbb{E}_{\mathcal{B}}[\mathcal{L}_{\mathrm{WM}}(\theta;\mathcal{B})] &= \mathbb{E}_{\mathcal{B}}\left[-\frac{1}{N_{\mathcal{B}}}\sum_{(h_t,y_t,o_{t+1})\in\mathcal{B}}\sum_{j=1}^{|o_{t+1}|}\log p_\theta(o_{t+1,j}\mid h_t, y_t, o_{t+1,<j})\right] \\ &= \mathbb{E}_Q[-\log p_\theta(Y\mid S)].\end{aligned} \tag{7}$$

Here the scoring inputs on the middle expression are understood after the recorded truncation. Selecting an example uniformly and then selecting one of its tokens uniformly would instead give an example-weighted objective. The construction above also preserves the random batch denominator: it does not replace an expectation of a ratio by a ratio of expectations.

Table 13: **Observation prediction on fixed transition pools.** Observation NLL is the total negative log-likelihood divided by the number of observation target tokens. Each environment uses the same examples and tokenization across its available checkpoints. The initial model is the reference; the other checkpoints use within-task or cross-task observation targets.

Environment	Checkpoint	Examples	Tasks	Target tokens	NLL
ALFWorld	Initial model	970	17	23,600	1.4269
	Cross-task target	970	17	23,600	0.9196
WebShop	Initial model	434	4	116,204	0.8916
	Within-task target	434	4	116,204	0.7156
	Cross-task target	434	4	116,204	0.6279
Search	Initial model	75	4	37,470	1.4028
	Within-task target	75	4	37,470	1.2263
	Cross-task target	75	4	37,470	1.2171

Proof of (5). Assume finite expected log loss and a normalized predictive distribution with positive mass on the support of Q. Adding and subtracting $\log Q(Y \mid S)$ gives

$$\begin{aligned}\mathbb{E}_Q[-\log p_\theta(Y \mid S)] &= \mathbb{E}_Q[-\log Q(Y \mid S)] + \mathbb{E}_Q\left[\log \frac{Q(Y \mid S)}{p_\theta(Y \mid S)}\right] \\ &= H_Q(Y \mid S) + \mathbb{E}_{S\sim Q} D_{\mathrm{KL}}(Q(\cdot \mid S) \| p_\theta(\cdot \mid S)). \end{aligned} \tag{8}$$

KL divergence is nonnegative, with equality precisely when the conditional distributions agree Q-almost surely. Hence an unrestricted predictor minimizes this risk at $p(\cdot \mid S) = Q(\cdot \mid S)$, the strict propriety of the logarithmic score (Gneiting & Raftery, 2007). The decomposition also holds conditionally on a fixed batch, with its empirical token law. A finite model need not attain the Bayes optimum, and empirical fitting alone does not establish held-out prediction quality.

The identified law is the *retained-token* law under the sampling and preprocessing rules. Variable lengths, target caps, and shortened histories prevent replacing it without further assumptions by a full-observation transition kernel. Likewise, changing the rollout policy changes Q; (5) holds with Q frozen during differentiation.

Information in the conditioning input. Suppose the retained input can be partitioned as $S = (V, U)$, where U is the retained generated-response information being tested and V contains the remaining input, including preceding target tokens. Over all normalized conditional predictors, define the full and reduced Bayes log risks as $R^*_{V,U} = \inf_p \mathbb{E}_Q[-\log p(Y \mid V, U)]$ and $R^*_V = \inf_p \mathbb{E}_Q[-\log p(Y \mid V)]$. The same decomposition yields

$$R^*_V - R^*_{V,U} = H_Q(Y \mid V) - H_Q(Y \mid V, U) = I_Q(Y; U \mid V) \geq 0, \tag{9}$$

where I_Q is conditional mutual information. This is an information identity for matched token distributions and unrestricted Bayes predictors. It establishes (4) in the main text. Finite trained response-conditioned observation predictors need not attain either infimum; changing preprocessing between arms would also change the comparison. The identity therefore explains the predictive question behind conditioning ablations, not a direct estimator of causal action effects or policy improvement.

D.2 Compatible Collaborative Updates

We use local auxiliary-learning analysis (Du et al., 2018; Yu et al., 2020) to characterize the combined gradient in (6). Fix a batch and write $F = \mathcal{L}_{\mathrm{PO}}$ and $G = \mathcal{L}_{\mathrm{WM}}$. All norms below are Euclidean. Let $g = \nabla F(\theta)$, $m = \nabla G(\theta)$, $\gamma = g^\top m$, and $d = g + \lambda m$, with $\lambda > 0$. Assume F and G are differentiable in an open convex neighborhood containing all points and line segments being compared, and their gradients are Lipschitz there with constants $\beta_F, \beta_G > 0$.

Proposition 1 (Local collaboration)**.** *Under these assumptions, the gradient step $\theta^+ = \theta - \eta d$, with $\eta > 0$, satisfies*

$$\begin{aligned} F(\theta^+) - F(\theta) &\leq -\eta(\|g\|^2 + \lambda\gamma) + \tfrac{\beta_F \eta^2}{2}\|d\|^2, \\ G(\theta^+) - G(\theta) &\leq -\eta(\gamma + \lambda\|m\|^2) + \tfrac{\beta_G \eta^2}{2}\|d\|^2. \end{aligned} \tag{10}$$

Consequently, F and G strictly decrease for a sufficiently small step when $\|g\|^2 + \lambda\gamma > 0$ and $\gamma + \lambda\|m\|^2 > 0$.

Proof of Proposition 1. For any displacement v whose segment $\{\theta + tv : 0 \leq t \leq 1\}$ lies in the neighborhood, gradient Lipschitzness implies the descent lemma

$$F(\theta + v) \leq F(\theta) + g^\top v + \tfrac{\beta_F}{2}\|v\|^2. \tag{11}$$

Indeed,

$$F(\theta + v) - F(\theta) - g^\top v = \int_0^1 [\nabla F(\theta + tv) - g]^\top v \, dt,$$

whose integrand is at most $\beta_F t\|v\|^2$. Substitute $v = -\eta d$ and use $g^\top d = \|g\|^2 + \lambda\gamma$. Applying the same argument to G, with $m^\top d = \gamma + \lambda\|m\|^2$, proves (10). If $g^\top d$ and $m^\top d$ are positive, then $d \neq 0$, and any step within the neighborhood satisfying

$$0 < \eta < \frac{2}{\|d\|^2} \min\left\{ \frac{\|g\|^2 + \lambda\gamma}{\beta_F}, \frac{\gamma + \lambda\|m\|^2}{\beta_G} \right\} \tag{12}$$

makes the upper bounds on the changes in F and G strictly negative. □

For nonzero gradients with $\gamma \geq 0$, $g^\top d$ and $m^\top d$ are positive for every $\lambda > 0$. If $\gamma < 0$, the two strict conditions reduce to

$$\frac{-\gamma}{\|m\|^2} < \lambda < \frac{\|g\|^2}{-\gamma}. \tag{13}$$

This interval is nonempty exactly when $\gamma^2 < \|g\|^2\|m\|^2$, namely when the nonzero gradients are not perfectly opposed. These are conditions for strict *first-order* descent of F and G; if an inner product is zero, curvature can determine the change and the proposition does not classify it.

D.3 Additional Progress Relative to Policy-Only Learning

Simultaneous descent and additional policy surrogate loss reduction are different claims. Compare two steps of the same size η from the same θ: $\theta_{\mathrm{PO}} = \theta - \eta g$ and $\theta^+ = \theta_{\mathrm{PO}} - \eta\lambda m$. Their local policy surrogate loss difference is

$$F(\theta^+) - F(\theta_{\mathrm{PO}}) = -\eta\lambda\gamma + O(\eta^2). \tag{14}$$

The Lipschitz-gradient assumption on F gives the two-sided remainder bound

$$F(\theta^+) - F(\theta_{\mathrm{PO}}) = -\eta\lambda\gamma + \varepsilon_\eta, \qquad |\varepsilon_\eta| \leq \beta_F\eta^2\lambda\left(\|g\|\|m\| + \tfrac{\lambda}{2}\|m\|^2\right). \tag{15}$$

Proof. Expand F at θ_{PO} along $-\eta\lambda m$. The absolute Taylor remainder is at most $\beta_F\eta^2\lambda^2\|m\|^2/2$. Replacing $\nabla F(\theta_{\mathrm{PO}})$ in the linear term by g incurs an additional absolute error of at most $\eta\lambda\|m\|\,\|\nabla F(\theta_{\mathrm{PO}}) - g\| \leq \beta_F\eta^2\lambda\|g\|\|m\|$. Adding the two bounds proves the claim and (14) for fixed θ and λ. □

In particular, the sufficient finite-step condition

$$\gamma > \beta_F\eta\left(\|g\|\|m\| + \tfrac{\lambda}{2}\|m\|^2\right) \tag{16}$$

implies $F(\theta^+) < F(\theta_{\mathrm{PO}})$. Positive overlap thus certifies additional policy surrogate loss reduction for a sufficiently small step. Negative overlap instead gives less policy surrogate loss reduction than the policy-only step for sufficiently small η, even if F and G decrease. Zero overlap leaves a second-order comparison.

The bound also explains why a fixed loss coefficient does not fix the strength of coupling. For $g, m \neq 0$, define $\rho = \gamma/(\|g\|\|m\|)$ and $r = \lambda\|m\|/\|g\|$. Then

$$g^\top d = \|g\|^2(1 + r\rho), \qquad \|d\|^2 = \|g\|^2(1 + 2r\rho + r^2). \tag{17}$$

The directional benefit and the curvature cost depend on the relative gradient magnitude r, as well as gradient cosine similarity ρ. This links the local analysis to the two gradient diagnostics in Appendix E, without inferring positive overlap from the observed norm ratios.

D.4 Optimizer Scope

Frozen updates and clipping. The statements concern fixed-batch losses, with advantages, rollout likelihoods, masks, and normalizers held constant. They do not differentiate through the sampling policy. A clipped GRPO surrogate is piecewise smooth; the smoothness argument applies within a neighborhood that does not cross a nondifferentiable clipping boundary. Repeated minibatch updates and refreshed on-policy data require a new local analysis, rather than treating the entire training run as descent on one stationary objective.

Preconditioning. The same proof covers a fixed symmetric positive-definite preconditioner M for the update $\theta^+ = \theta - \eta M(g + \lambda m)$. Apply the Euclidean result to $v \mapsto F(\theta + M^{1/2}v)$ and $v \mapsto G(\theta + M^{1/2}v)$. The relevant gradients become $M^{1/2}g, M^{1/2}m$, so overlap is $g^\top Mm$, and smoothness is measured in these transformed coordinates. Adam's momentum and loss-dependent second moments are not a fixed common preconditioner for two counterfactual updates. The proposition is therefore a local analysis of the combined objective, not an assertion that raw-gradient cosine similarity determines an actual Adam step. The relative-step comparison also matches step size, not wall-clock compute.

E GRADIENT DIAGNOSTICS

E.1 OBSERVATION TARGETS AND MODEL DEPTH CHANGE GRADIENT STRENGTH

Equal coefficients produce different auxiliary gradient strengths. At each of three checkpoints from the ALFWorld within-task run, we hold its model and freshly sampled diagnostic batch fixed across target constructions and compute policy and auxiliary gradients. At update 150, the coefficient-weighted auxiliary-to-policy norm ratio is 0.2418 for correctly paired observations, 0.0526 for within-task observations, and 0.7499 for cross-task observations. With the same coefficient $\lambda = 0.1$, the cross-task target produces 14.25 times the auxiliary gradient norm of the within-task target. The separation in relative magnitude is visible at all three checkpoints (Fig. 12, top). Target construction therefore changes the strength of observation supervision in the combined gradient, even when the nominal coefficient is fixed.

The prediction losses give context to this scale difference. At update 150, the within-task and cross-task targets have diagnostic NLLs of 0.3062 and 0.9137, respectively. In a model trained with within-task supervision, cross-task reassignment thus produces both a larger prediction loss and a larger auxiliary gradient. The clearest measured difference is in magnitude: the policy–auxiliary cosines remain near zero, with absolute values at most 0.033. These measurements characterize the gradients before an optimizer update. The following subsections give the full diagnostic values, donor construction, and numerical sensitivity.

The auxiliary contribution also varies across model depth. We measure correctly paired observation-prediction gradients at the initial models in all three environments, using the native policy surrogate with its invalid-response penalty. The coefficient-weighted auxiliary-to-policy norm ratio is higher in the deep than middle transformer-layer group: 8.83 versus 2.20 on ALFWorld, 0.70 versus 0.21 on WebShop, and 0.93 versus 0.51 on Search (Fig. 12, bottom). These ratios use concatenated gradients within each depth group. The repeated pattern shows that observation supervision makes a relatively stronger contribution to the deep-layer gradient than to the middle-layer gradient across these environments. Together with the stage measurements, the depth analysis reveals how the additional objective contributes at different parts of the shared model and how its relative strength depends on the observation targets.

E.2 DEFINITIONS AND TRAINING-STAGE MEASUREMENTS

Let g be the policy-surrogate-loss gradient and $u = \lambda \nabla \mathcal{L}_{\mathrm{WM}}$ the coefficient-weighted auxiliary gradient, with $\lambda = 0.1$ applied once. We report their cosine $g^\top u/(\|g\|\|u\|)$, norm ratio $\|u\|/\|g\|$, and relative projection $g^\top u/\|g\|^2$. The latter equals the product of cosine and norm ratio. We concatenate parameter gradients before computing these statistics; layer-group cosines are not averages of individual layer cosines.

The stage study uses the ALFWorld within-task checkpoints at updates 50, 100, and 150. At each checkpoint, we collect eight fresh on-policy trajectories for each of the same 16 tasks, then hold the resulting batch and model parameters fixed. The three auxiliary constructions preserve the target and target-length multisets and have zero fallback. This diagnostic's cross-task donor must come from a different semantic task type, a stricter rule than the task-identity constraint used in cross-task training. Across checkpoints, the model and its sampled trajectories change.

We compute native policy-surrogate-loss and dense auxiliary cross-entropy gradients without an optimizer update, using padded inputs, physical micro-batch size 4, logical micro-batch size 32, accumulation 4, and two workers. The reference model is the initial policy, while old-policy log probabilities of generated-response tokens are recomputed with the current actor. All parameter hashes are unchanged after measurement. The update-50 within-task numerical calibration gives a maximum global relative gradient L2 difference of 2.03% across the tested padding conditions. The protocol marks cosine values within ± 0.06 as directionally indeterminate. This empirical sensitivity range is not a confidence interval or a separately established error bound for the later checkpoints. Tab. 14 reports the full values; its NLL uses each stage's diagnostic batch, distinct from the held-out losses in Appendix C.2.

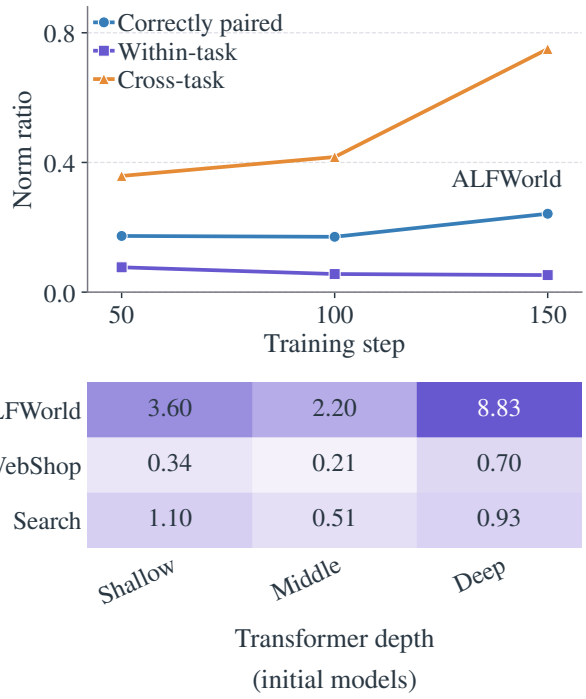

Figure 12: Top: coefficient-weighted auxiliary-to-policy gradient norm ratios at ALFWorld within-task checkpoints. Bottom: corresponding ratios across shallow, middle, and deep transformer layers, measured on initial models with correctly paired supervision and native policy surrogates in each environment. Both diagnostics use $\lambda = 0.1$; heatmap cells report ratios with logarithmic color shading.

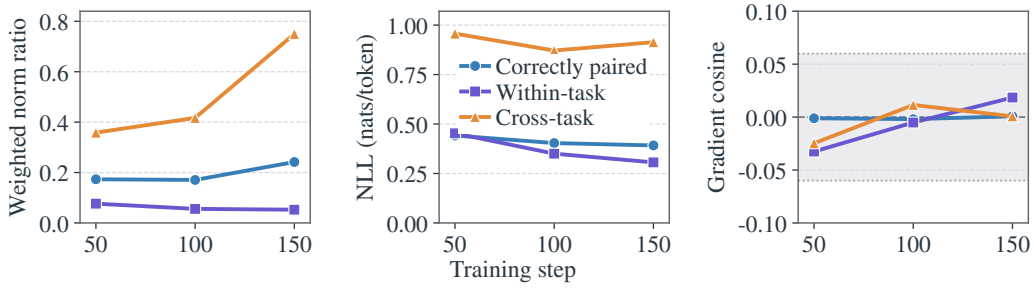

Figure 13: At each checkpoint of the ALFWorld within-task run, all three targets share the model and diagnostic batch. Panels show $\|\lambda\nabla\mathcal{L}_{\mathrm{WM}}\|/\|\nabla\mathcal{L}_{\mathrm{PO}}\|$ at $\lambda = 0.1$, target-token NLL, and policy–auxiliary gradient cosine. Gray shading marks the empirical numerical-sensitivity band $[-0.06, 0.06]$.

E.3 Reward-Source and Depth Diagnostics

The reward-source study uses initial Qwen2.5-1.5B models for ALFWorld and WebShop and the initial 3B model for Search. For each environment, the original captured batch is fixed across three policy-surrogate-loss definitions. Each batch contains 128 trajectories from 16 tasks. We retain the saved old-policy log probabilities of generated-response tokens, normalize the policy surrogate loss over generated tokens, and normalize the auxiliary loss over target tokens. *Task* uses task reward alone. *Native* adds the invalid-response penalty and recomputes GRPO advantages. *Full* includes the explicit entropy and KL terms. Advantage normalization is recomputed for Task and Native, so their gradient difference is not an independently additive penalty component.

Table 14: **Gradient scale varies across observation-target pairings and checkpoints.** Frozen diagnostics at ALFWorld within-task checkpoints. The norm ratio and projection include the auxiliary coefficient $\lambda = 0.1$; observation NLL is in nats per target token on each checkpoint's diagnostic batch.

Checkpoint	Target pairing	Cosine	Norm ratio	Projection	NLL
50	Correctly paired	−0.0011	0.1733	−0.00019	0.4419
	Within-task	−0.0325	0.0766	−0.00249	0.4532
	Cross-task	−0.0247	0.3584	−0.00886	0.9570
100	Correctly paired	−0.0019	0.1707	−0.00033	0.4037
	Within-task	−0.0051	0.0558	−0.00028	0.3502
	Cross-task	+0.0115	0.4168	+0.00478	0.8713
150	Correctly paired	+0.0008	0.2418	+0.00018	0.3918
	Within-task	+0.0185	0.0526	+0.00097	0.3062
	Cross-task	+0.0007	0.7499	+0.00054	0.9137

Table 15: **The policy surrogate loss changes the relative auxiliary gradient.** Initial-model, fixed-batch diagnostics. Task uses task reward alone; Native includes the invalid-response penalty; Full adds the explicit regularizers. The three columns use correctly paired, within-task, and cross-task observation targets.

		Cosine			Weighted auxiliary / policy norm		
Environment	Policy loss	Correctly paired	Within-task	Cross-task	Correctly paired	Within-task	Cross-task
ALFWorld	Task	−0.0160	−0.0164	−0.0167	47.0166	46.9399	46.8250
	Native	−0.0092	−0.0111	−0.0095	3.8620	3.8557	3.8462
	Full	−0.0093	−0.0111	−0.0096	3.8617	3.8554	3.8459
WebShop	Task	−0.0212	−0.0204	−0.0196	2.6102	2.7483	2.8392
	Native	+0.0144	+0.0154	+0.0160	0.3576	0.3766	0.3890
	Full	+0.0145	+0.0156	+0.0162	0.3570	0.3759	0.3883
Search	Task	−0.0555	−0.0578	−0.0639	1.3195	1.3424	1.3869
	Native	−0.0396	−0.0408	−0.0440	0.8627	0.8776	0.9067
	Full	−0.0396	−0.0408	−0.0440	0.8627	0.8776	0.9067

The invalid-response penalty, entropy coefficient, and KL coefficient are 0.1, 0.001, and 0.01 for ALFWorld/WebShop, and 0.01, 0, and 0.001 for Search. In this initial-model diagnostic, reference log probabilities equal the detached current-model values, giving zero KL value and gradient. Search therefore reuses Native as Full because its entropy coefficient is also zero. The Task/Native/Full geometry is reported in Tab. 15; the policy-surrogate-loss gradient direction changes and coefficient-weighted entropy norms are reported in Tab. 16.

The depth summary in Tab. 17 uses shallow, middle, and deep transformer-layer groups: layers 0–9, 10–18, and 19–27 for the 1.5B model, and 0–11, 12–23, and 24–35 for the 3B model. The shared embedding/output and final normalization parameters are included in the global statistic and kept separate from these three transformer-layer groups. Supervision with correctly paired observation targets has a larger relative norm in the deep than middle group in all three environments: 8.83 versus 2.20, 0.70 versus 0.21, and 0.93 versus 0.51. Thus the global ratio summarizes a coupling that also varies across model depth. All measurements hold parameters fixed and characterize the objectives' gradients before the optimizer step.

Table 16: **Reward construction changes both policy scale and direction.** The last column compares the coefficient-weighted entropy gradient with the Native gradient. Search has zero entropy coefficient.

	Policy gradient norm		Native / Task	Cosine	Entropy / Native
Environment	Task	Native	norm	(Task, Native)	norm
ALFWorld	0.1927	2.3455	12.1743	-0.1901	0.00816
WebShop	0.2772	2.0232	7.2986	$+0.1572$	0.00986
Search	0.9462	1.4473	1.5296	$+0.6979$	0

Table 17: **The coupling also varies across model depth.** Correctly paired observation-prediction gradients relative to the Native policy surrogate loss at the initial model. Depth-group statistics are computed from concatenated gradients, not by averaging layer-level cosines.

	Cosine			Weighted auxiliary / policy norm		
Environment	Shallow	Middle	Deep	Shallow	Middle	Deep
ALFWorld	-0.0044	-0.0276	$+0.0005$	3.5992	2.1980	8.8320
WebShop	$+0.0157$	$+0.0251$	$+0.0193$	0.3443	0.2095	0.7046
Search	-0.0596	-0.0372	-0.0130	1.1037	0.5087	0.9313

F Behavior on Shared Evaluation Tasks

F.1 Evaluation and Paired Statistics

We reevaluate available checkpoints on the same tasks within each environment. ALFWorld includes the initial model, policy-only optimization, and cross-task supervision; WebShop includes the initial model, policy-only optimization, within-task supervision, and cross-task supervision; Search includes the initial model, within-task supervision, and cross-task supervision. The initial model is the instruction-tuned checkpoint, and policy-only optimization is GRPO without an auxiliary objective. ALFWorld and WebShop use the 1.5B model, and Search uses the 3B model. The initial model is evaluated before training; the remaining checkpoints are evaluated after 150 updates for ALFWorld and WebShop and 200 updates for Search. The common-task reevaluations form a separate evaluation campaign from the historical training endpoints in Appendix B; each result table identifies the campaign.

ALFWorld and WebShop each contain 128 episodes per checkpoint, covering 91 and 126 distinct task identities, respectively. ALFWorld task identities come from the recorded game files. WebShop goal identities are reconstructed from the seeded goal sequence and verified against the saved instructions and goal catalog. Search contains 51,713 episodes covering 50,508 distinct questions across seven datasets. Within each environment, every checkpoint uses the same task identities, repetition counts, and evaluation settings. The evaluation uses environment seed 0.

Success follows each environment's task criterion. The mean number of interaction turns per episode includes successful and failed episodes. Format errors count failures of the native action projector, divided by all recorded actions. Tab. 19 reports the underlying counts and recorded failure endpoints. Environment-terminal failures and collector-horizon failures identify distinct recorded stopping conditions; an environment-terminal event may itself include an internal step limit.

We use 5,000 paired task-cluster bootstrap draws. Each draw resamples task identities and retains all episodes associated with a sampled identity. Search resampling is stratified by dataset; Avg gives each dataset equal weight. Unadjusted, pointwise 95% percentile intervals quantify evaluation-task variation for the fixed checkpoints. Every comparison is written in subtraction order.

Table 18: **Common-task reevaluation separates success from interaction turns.** Each model completes 128 episodes per environment. Mean interaction turns per episode include successful and failed episodes; format errors are native action-projector failures as a percentage of recorded actions. Policy-only denotes GRPO without auxiliary supervision.

Model	Success (%)	Mean turns	Format errors (%)
ALFWorld			
Instruction-tuned base	1.56	49.64	11.36
Policy-only	57.81	28.52	0.74
Cross-task	65.62	27.99	0.08
WebShop			
Instruction-tuned base	3.91	12.32	90.30
Policy-only	55.47	8.73	0.18
Within-task	60.94	6.15	1.40
Cross-task	61.72	7.55	0.21

F.2 ALFWorld and WebShop Checkpoint Comparisons

Tab. 20 reports all three ALFWorld and six WebShop pairs among the checkpoints listed above. On WebShop, within-task supervision uses fewer interaction turns per episode than policy-only optimization and cross-task supervision, while the success-rate intervals for these comparisons include zero. The common-task measurements therefore distinguish changes in interaction turns from changes in success rate.

Table 19: **Episode outcomes and recorded failure endpoints.** Invalid/actions counts format failures among recorded actions. Terminal failures and collector-horizon failures are separate native labels; an environment terminal can include an internal step limit. Search success counts are pooled counts, not the equal-weight mean of the seven dataset success rates (Avg).

Model	Success/episodes	Invalid/actions	Failures	Terminal	Horizon
ALFWorld					
Instruction-tuned base	2/128	722/6,354	126	126	0
Policy-only	74/128	27/3,650	54	54	0
Cross-task	84/128	3/3,583	44	44	0
WebShop					
Instruction-tuned base	5/128	1,424/1,577	123	41	82
Policy-only	71/128	2/1,117	57	40	17
Within-task	78/128	11/787	50	49	1
Cross-task	79/128	2/966	49	42	7
Search					
Instruction-tuned base	15,655/51,713	38,341/120,654	36,058	36,058	0
Within-task	23,434/51,713	46/141,575	28,279	28,279	0
Cross-task	22,442/51,713	51/159,518	29,271	29,271	0

Table 20: **Paired comparisons among the reported ALFWorld and WebShop checkpoints.** Each cell gives the indicated difference and its 95% task-cluster bootstrap interval. Success and format-error differences are absolute rate differences (%); turn differences compare mean interaction turns per episode.

Comparison	Δ success (%)	Δ mean turns	Δ format errors (%)
ALFWorld (91 task clusters)			
Policy-only − Instruction-tuned base	+56.25 [+45.60, +66.67]	−21.12 [−25.36, −16.93]	−10.62 [−12.71, −8.73]
Cross-task − Instruction-tuned base	+64.06 [+54.33, +74.07]	−21.65 [−25.77, −17.69]	−11.28 [−13.34, −9.42]
Cross-task − Policy-only	+7.81 [−0.82, +17.21]	−0.52 [−3.87, +2.69]	−0.66 [−0.99, −0.35]
WebShop (126 task clusters)			
Policy-only − Instruction-tuned base	+51.56 [+42.30, +60.32]	−3.59 [−4.53, −2.64]	−90.12 [−90.83, −89.35]
Within-task − Instruction-tuned base	+57.03 [+48.03, +65.62]	−6.17 [−6.94, −5.39]	−88.90 [−90.01, −87.76]
Cross-task − Instruction-tuned base	+57.81 [+48.82, +66.41]	−4.77 [−5.65, −3.81]	−90.09 [−90.86, −89.29]
Within-task − Policy-only	+5.47 [−3.10, +14.17]	−2.58 [−3.27, −1.89]	+1.22 [+0.37, +2.16]
Cross-task − Policy-only	+6.25 [−1.54, +14.06]	−1.18 [−1.94, −0.45]	+0.03 [−0.35, +0.42]
Cross-task − Within-task	+0.78 [−6.25, +7.87]	+1.40 [+0.80, +1.98]	−1.19 [−2.10, −0.37]

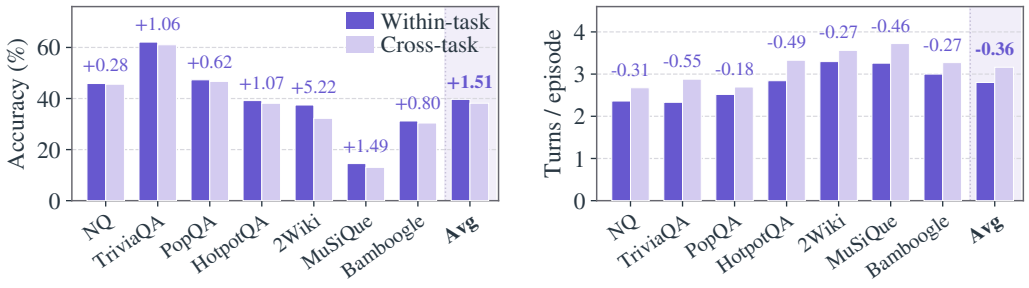

Figure 14: Bars show common-task Search checkpoint results; labels give within-task minus cross-task differences (absolute accuracy differences in %). Avg equally weights the seven datasets; turn means include all episodes. The absolute Avg accuracy difference is 1.51% (95% paired interval [0.68%, 2.29%]); all paired intervals are in Appendix F.

F.3 Search Dataset Outcomes and Paired Comparisons

Tab. 21 reports all seven dataset success rates and Avg. Tab. 22 adds mean interaction turns per episode and format errors. Relative to the initial model, within-task and cross-task supervision yield absolute Avg improvements of 14.23% and 12.72%, respectively, and use more interaction turns per episode on every dataset. Comparing the auxiliary targets, within-task supervision yields an absolute Avg improvement of 1.51% over cross-task supervision while using fewer interaction turns per episode on every dataset.

The three pairwise comparisons are reported in Tab. 23, Tab. 24, and Tab. 25. The largest success-rate difference between within-task and cross-task supervision is on 2WikiMultiHopQA, at 5.22% in absolute terms. The success intervals favor within-task supervision on five datasets; the NQ and Bamboogle intervals include zero. Turn differences and their intervals are reported by dataset.

Table 21: **Search common-task success across all seven datasets.** Rates are computed from the complete per-question records. Avg is the equal-weight mean of the seven success rates, computed before rounding.

Dataset	Episodes	Question clusters	Instruction-tuned base (%)	Within-task (%)	Cross-task (%)
NQ	3,610	3,610	22.71	45.87	45.60
TriviaQA	11,313	11,312	47.86	62.10	61.04
PopQA	14,267	13,068	29.11	47.32	46.70
HotpotQA	7,405	7,405	26.20	39.19	38.12
2WikiMultiHopQA	12,576	12,576	24.86	37.45	32.23
MuSiQue	2,417	2,412	7.32	14.52	13.03
Bamboogle	125	125	20.00	31.20	30.40
Avg	–	–	25.44	39.66	38.16

F.4 New and Lost Successes on Paired Episodes

Paired outcomes identify changes in which episodes succeed. We pair each recorded episode with the corresponding episode in the other checkpoint's evaluation, preserving task identity and its position in the shared evaluation sequence. Repeated task identities retain their separate episode slots. For each comparison, a new success means that the reference fails and the comparison succeeds; a lost success means that the reference succeeds and the comparison fails. Shared successes and shared failures complete the four mutually exclusive outcomes.

Tab. 26 decomposes the WebShop comparisons. The within-task–cross-task difference combines 10 new and 11 lost successes. Task-cluster intervals for the net success differences appear in Tab. 20.

Table 22: **The within-task target uses fewer interaction turns per episode than the cross-task target on every Search dataset.** Mean interaction turns include every episode; format errors use the native action projector.

	Mean turns			Format errors (%)		
Dataset	Instruction-tuned base	Within-task	Cross-task	Instruction-tuned base	Within-task	Cross-task
NQ	2.085	2.361	2.676	27.554	0.035	0.062
TriviaQA	1.783	2.331	2.880	29.317	0.068	0.052
PopQA	2.278	2.518	2.694	30.809	0.022	0.008
HotpotQA	2.456	2.843	3.328	34.809	0.033	0.016
2WikiMultiHopQA	2.812	3.296	3.562	33.655	0.010	0.025
MuSiQue	2.746	3.259	3.724	30.511	0.076	0.089
Bamboogle	2.120	3.000	3.272	30.943	0.000	0.489

Table 23: **Search paired comparison: Within-task − Instruction-tuned base.** Differences and pointwise 95% question-cluster intervals; success and format errors are absolute rate differences (%). Turn differences compare mean interaction turns per episode. Avg weights the seven datasets equally.

Dataset	Δ success (%)	Δ mean turns	Δ format errors (%)
NQ	+23.16 [+21.55, +24.76]	+0.28 [+0.24, +0.31]	−27.52 [−28.05, −26.98]
TriviaQA	+14.24 [+13.40, +15.06]	+0.55 [+0.53, +0.57]	−29.25 [−29.62, −28.86]
PopQA	+18.21 [+17.43, +19.01]	+0.24 [+0.22, +0.26]	−30.79 [−31.09, −30.47]
HotpotQA	+12.99 [+11.94, +14.00]	+0.39 [+0.36, +0.41]	−34.78 [−35.13, −34.43]
2WikiMultiHopQA	+12.60 [+11.69, +13.49]	+0.48 [+0.47, +0.50]	−33.65 [−33.89, −33.41]
MuSiQue	+7.20 [+5.80, +8.58]	+0.51 [+0.47, +0.56]	−30.43 [−30.97, −29.92]
Bamboogle	+11.20 [+2.40, +20.00]	+0.88 [+0.67, +1.07]	−30.94 [−33.93, −28.03]
Avg	+14.23 [+12.86, +15.53]	–	–

Table 24: **Search paired comparison: Cross-task − Instruction-tuned base.** Difference and interval definitions follow Table 23.

Dataset	Δ success (%)	Δ mean turns	Δ format errors (%)
NQ	+22.88 [+21.27, +24.46]	+0.59 [+0.55, +0.63]	−27.49 [−28.03, −26.94]
TriviaQA	+13.18 [+12.34, +14.01]	+1.10 [+1.08, +1.11]	−29.27 [−29.64, −28.88]
PopQA	+17.59 [+16.80, +18.39]	+0.42 [+0.40, +0.44]	−30.80 [−31.10, −30.49]
HotpotQA	+11.92 [+10.88, +12.96]	+0.87 [+0.85, +0.90]	−34.79 [−35.14, −34.45]
2WikiMultiHopQA	+7.37 [+6.46, +8.32]	+0.75 [+0.73, +0.77]	−33.63 [−33.87, −33.40]
MuSiQue	+5.71 [+4.42, +7.04]	+0.98 [+0.93, +1.02]	−30.42 [−30.97, −29.90]
Bamboogle	+10.40 [+1.60, +19.20]	+1.15 [+0.95, +1.34]	−30.45 [−33.58, −27.32]
Avg	+12.72 [+11.41, +14.03]	–	–

Table 25: **Search paired comparison: Within-task − Cross-task.** Difference and interval definitions follow Table 23.

Dataset	Δ success (%)	Δ mean turns	Δ format errors (%)
NQ	$+0.28$ $[-0.78, +1.30]$	-0.31 $[-0.33, -0.30]$	-0.03 $[-0.10, +0.05]$
TriviaQA	$+1.06$ $[+0.50, +1.60]$	-0.55 $[-0.56, -0.54]$	$+0.02$ $[-0.03, +0.06]$
PopQA	$+0.62$ $[+0.15, +1.08]$	-0.18 $[-0.19, -0.17]$	$+0.01$ $[-0.00, +0.04]$
HotpotQA	$+1.07$ $[+0.22, +1.90]$	-0.49 $[-0.50, -0.47]$	$+0.02$ $[-0.01, +0.05]$
2WikiMultiHopQA	$+5.22$ $[+4.41, +6.04]$	-0.27 $[-0.28, -0.26]$	-0.01 $[-0.03, +0.00]$
MuSiQue	$+1.49$ $[+0.25, +2.73]$	-0.46 $[-0.49, -0.44]$	-0.01 $[-0.11, +0.08]$
Bamboogle	$+0.80$ $[-4.80, +5.60]$	-0.27 $[-0.38, -0.17]$	-0.49 $[-1.49, -0.00]$
Avg	$+1.51$ $[+0.68, +2.29]$	–	–

Table 26: **WebShop net success differences combine new and lost successes.** Counts use the same 128 recorded episodes for each comparison. The arrow points from the reference to the comparison checkpoint; new/lost successes use that direction.

Environment	Reference → comparison	New	Lost	Shared success	Shared failure
WebShop	Policy-only → Within-task	19	12	59	38
WebShop	Policy-only → Cross-task	17	9	62	40
WebShop	Cross-task → Within-task	10	11	68	39

For Search, Tab. 27 decomposes within-task supervision relative to cross-task supervision separately on each dataset. The difference in average accuracy is computed by subtracting the lost-success rate from the new-success rate within each dataset and averaging the seven differences. Pooling the episode counts would give a different statistic because dataset sizes differ.

Table 27: **Search success turnover, with the cross-task target as reference and the within-task target as comparison.** Dataset rows are episode counts. The final row gives the equally weighted mean of seven dataset rates (%). Before rounding, subtracting lost from new success gives the absolute Avg difference of $+1.51\%$.

Dataset	Episodes	New	Lost	Shared success	Shared failure
NQ	3,610	192	182	1,464	1,772
TriviaQA	11,313	576	456	6,449	3,832
PopQA	14,267	586	497	6,165	7,019
HotpotQA	7,405	554	475	2,348	4,028
2WikiMultiHopQA	12,576	1,761	1,104	2,949	6,762
MuSiQue	2,417	131	95	220	1,971
Bamboogle	125	6	5	33	81
Avg rate (%)	–	6.60	5.10	33.06	55.24

G WorldPO Training Procedure and Implementation

G.1 Training and Evaluation Configurations

Tab. 28 summarizes benchmark-specific settings recovered from the WorldPO training records. ALFWorld and WebShop combine SFT-based observation prediction with the policy optimizer named in Tab. 1; Search uses GRPO + SFT. Published Search baselines retain the settings of their original studies.

Table 28: **WorldPO training and evaluation configurations.** Lengths are measured in tokens; the training batch counts tasks before sampling grouped trajectories. Model sizes refer to Qwen2.5-Instruct. The table summarizes source-verified runs; optimizer-specific and resource variations are explained below.

Hyperparameter	ALFWorld	WebShop	Search 3B	Search 7B
Model size	1.5B / 3B	1.5B / 3B	3B	7B
Learning				
Training task batch	16	16	256	256
Trajectories per task	8	8	5	5
Policy-update minibatch	256	64	512	512
Training updates	150	150	200	200
Learning rate	10^{-6}	10^{-6}	10^{-6}	10^{-6}
Weight decay	0.01	0.01	0.01	0.01
Warmup ratio	0	0	0.1	0.1
Learning-rate schedule	Constant	Constant	Constant	Constant
Policy epochs per update	1	1	1	1
Training temperature	1.0	1.0	1.0	1.0
Interaction and policy objective				
History window (turns)	2	2	4	4
Maximum interaction turns	50	15	4	4
Maximum prompt length	2048	4096	4096	4096
Maximum response length	512	512	512	512
Policy clipping ratio	0.2	0.2	0.2	0.2
Gradient-norm clipping	1.0	1.0	1.0	1.0
KL-loss coefficient	0.01	0.01	0.001	0.001
Entropy coefficient	0.001	0.001	0	0
Invalid-action penalty	0.1	0.1	0.01	0.01
Discount factor γ	1 / 0.95	1 / 0.95	0.95	0.95
World modeling				
Observation-prediction objective	SFT	SFT	SFT	SFT
Auxiliary coefficient λ	0.1	0.1	0.1	0.1
Maximum prediction sequence	2048	4096	2048	4096
Maximum observation target	512	512	512	512
Evaluation and parallelism				
Evaluation decoding	Sampling	Sampling	Greedy	Greedy
Evaluation temperature	0.4	0.4	0	0
Evaluation top-p	1.0	1.0	1.0	1.0
Evaluation top-k	Disabled	Disabled	Disabled	Disabled
Evaluation samples per task	1	1	1	1
Nodes / GPUs per node	1 / 1, 2, 4	1 / 1–2	1 / 2	1 / 4
Rollout tensor parallelism	1–2	1–2	1	1

GRPO and RLOO use $\gamma = 1$ on ALFWorld and WebShop; GiGPO uses $\gamma = 0.95$ and step-advantage weight 1. The 3B ALFWorld GiGPO baseline is evaluated offline at update 150 with environment

seed 0 and a 50-turn limit. GiGPO uses mean-and-standard-deviation advantage normalization on ALFWorld and mean normalization on WebShop. The learning rate is constant after the listed warmup; the KL penalty uses the low-variance estimator. The environment seed is 0. The recovered ALFWorld and WebShop evaluation configurations specify 128 episodes; evaluation batch size is 1 for WebShop GiGPO + SFT and 128 for the other recorded ALFWorld and WebShop runs. Search uses evaluation batches of 512. No fixed example-count cap is imposed on the auxiliary prediction pool. Single-GPU runs use rollout tensor parallelism 1; the recorded two-GPU ALFWorld and WebShop runs use tensor parallelism 2. The 3B ALFWorld GiGPO + SFT run resumes from update 61 on four GPUs with tensor parallelism 1.

G.2 Training Procedure

Algorithm 1 summarizes collaborative training. The policy optimizer supplies its own advantage estimates, credit assignment, and regularization. The prediction objective uses the observations collected by the same policy. Our default uses uniform observation-token supervision and $\lambda = 0.1$.

Algorithm 1 WorldPO: Collaborative policy optimization and world modeling

Require: Shared model θ, real environment $\mathcal{E}$, policy surrogate loss $\mathcal{L}_{\mathrm{PO}}$, coefficient λ

1: **while** training budget remains **do**
2: Sample tasks and collect grouped trajectories with the current policy.
3: Compute returns and advantages required by the selected policy surrogate loss.
4: Build $\mathcal{B}$ from observed transitions (h_t, y_t, o_{t+1}).
5: Accumulate policy and observation-prediction gradients on θ.
6: Update θ using the combined loss in (3).
7: **end while**
8: Deploy π_θ with the original interaction interface.

G.3 Causal Inputs and Prediction Targets

For each eligible interaction, the builder concatenates the agentic history h_t supplied to the policy with its current generated response y_t, including any generated reasoning and the executable action. The environment executes the extracted action a_t, whereas observation prediction conditions on the complete generated response before truncation. The environmental observation o_{t+1} provides the prediction target. Historical context is the context actually available to the policy, rather than an unrestricted reconstruction of the trajectory. The ALFWorld and WebShop configurations use a two-turn history window; Search uses a four-turn window and a four-turn interaction budget. Prior actions and observations are formatted by the environment manager. WebShop omits the historical context if its formatted input exceeds the manager's 13,000-character limit.

The builder caps the target length and removes the oldest prefix tokens as needed to satisfy the configured sequence limit. In (2), h_t, y_t, and o_{t+1} denote the history, generated-response, and observation token sequences retained after preprocessing and truncation; $|o_{t+1}|$ counts retained observation tokens. The target observation is not included in the conditioning history or response; preceding observation tokens enter only through teacher forcing. Prompts, earlier observations, generated reasoning, actions, and padding receive no auxiliary prediction loss.

G.4 Shared-Parameter Updates

The policy and predictor share parameters, $\phi = \theta$. The notation p_θ denotes observation prediction by the same LLM used for π_θ. Auxiliary micro-batches contribute to the same actor update as the policy surrogate loss, with the global target-token denominator preserved across distributed workers.

Observation prediction adds forward and backward computation to the policy update. The implementation supports packed causal inputs and vocabulary projection at supervised positions to reduce padding and unused-logit overhead while preserving the loss. The auxiliary objective changes training, while deployment retains the original policy interface.

H Extended Preliminaries

H.1 Interaction and Trajectory Distribution

For a task $x \sim \mathcal{D}$, the POMDP $\mathcal{E}$ starts with an initial observation o_0. At turn t, the policy generates $y_t \sim \pi_\theta(\cdot \mid h_t)$ from $h_t = (x, o_0, y_0, o_1, \ldots, y_{t-1}, o_t)$. A fixed parser extracts the executable action a_t from y_t; the environment executes a_t and returns o_{t+1}. The complete generated response remains available for later decisions and observation prediction.

Marginalizing the hidden state given the agentic history defines the initial-observation law $p_{\mathcal{E}}(o_0 \mid x)$ and conditional observation law $p_{\mathcal{E}}(o_{t+1} \mid h_t, a_t)$. Observations include terminal status, and an episode stops at termination or the finite turn budget. For a valid completed trajectory $\tau = (x, o_0, y_0, \ldots, y_{T-1}, o_T)$, the probability is

$$p_{\theta,\mathcal{E}}(\tau \mid x) = p_{\mathcal{E}}(o_0 \mid x) \prod_{t=0}^{T-1} \pi_\theta(y_t \mid h_t) p_{\mathcal{E}}(o_{t+1} \mid h_t, a_t). \tag{18}$$

The horizon T may vary across completed trajectories in (1). The fixed parser makes a_t determined by y_t, so conditioning on the complete generated response also specifies the executed action whose consequence is predicted.

H.2 Policy Objective and Gradient

Hold the task distribution, environment, parser, and return function fixed. Assuming differentiable policy probabilities with parameter-independent support and sufficient integrability to exchange differentiation and expectation, the likelihood-ratio identity applied to (1) gives

$$\begin{aligned} \nabla_\theta J(\theta) &= \mathbb{E}_{x,\tau}[r(\tau) \nabla_\theta \log p_{\theta,\mathcal{E}}(\tau \mid x)] \\ &= \mathbb{E}_{x,\tau}\left[r(\tau) \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(y_t \mid h_t) \right]. \end{aligned} \tag{19}$$

For a fixed trajectory, the environment factors in (18) have no policy-parameter derivatives. Observations still influence the policy gradient through the generated-response probabilities conditioned on the history.

H.3 World-Modeling Objectives

For likelihood-based world modeling, let μ be a distribution of collected records (h_t, y_t, o_{t+1}), held fixed during fitting. The expected sequence negative log-likelihood is

$$\mathcal{L}_{\mathrm{WM}}(\phi; \mu) = -\mathbb{E}_{(h_t, y_t, o_{t+1}) \sim \mu}[\log p_\phi(o_{t+1} \mid h_t, y_t)]. \tag{20}$$

The distribution μ may be induced by a collection policy and environment or given by the empirical distribution of a recorded dataset.

For a textual observation, the sequence likelihood factorizes as

$$p_\phi(o_{t+1} \mid h_t, y_t) = \prod_{j=1}^{|o_{t+1}|} p_\phi(o_{t+1,j} \mid h_t, y_t, o_{t+1,<j}). \tag{21}$$

Here $|o_{t+1}|$ is the observation's token length including its end marker, $o_{t+1,j}$ is token j, and $o_{t+1,<j}$ contains preceding target tokens. Teacher forcing supplies those preceding tokens when scoring the sequence. (20) averages sequence NLL over records. WorldPO's implementation instead averages over retained observation tokens, as specified in (2); input preprocessing is detailed in Appendix G.

Preference objectives over candidate observations and rewards for prediction quality provide alternatives to likelihood fitting (Wang et al., 2025b; Yu et al., 2026). Preference-based training requires candidate predictions for the same conditioning input and preference supervision reflecting their predictive quality. Fig. 3 illustrates alternative operators; WorldPO's implementation uses the observation-token loss in Sec. 3.1.

I CASE STUDIES OF AGENTIC REASONING

We compare WorldPO with policy-only optimization (GRPO) on the same evaluation tasks to examine how task outcomes emerge from sequences of executable actions and returned observations. These cases use correctly paired observation supervision, rather than the reassigned targets studied in Appendix F. We reevaluate the fixed 1.5B checkpoints after 150 updates on ALFWorld and WebShop, and the fixed 3B checkpoints after 200 updates on Search. The evaluations retain complete generated responses, extracted actions, and returned observations for 128 episodes per checkpoint on ALFWorld and WebShop and 51,713 on Search. WebShop cases use the original 15-turn horizon, not the separate extended-horizon diagnostic. ALFWorld pairs share the recorded initial state; WebShop goal identities are reconstructed from the seeded catalog and checked against the instructions; Search pairs share the original dataset row and question.

The same logs support the aggregate behavioral comparisons in Sec. 5.2. Execution-turn means include the terminal answer or purchase; Search means are weighted by episode count across the seven datasets. The common-success comparison retains only episodes solved by both methods, while the all-episode comparison also includes failures and horizon-limited trajectories. In ALFWorld, the invalid action rate is the fraction of all recorded action attempts returning "Nothing happens." GRPO incurs 939 invalid actions in 3,714 attempts (25.28%), compared with 378 in 3,102 for WorldPO (12.19%). The count covers every action category, including navigation, examination, and object manipulation. The 53 GRPO and one WorldPO format-invalid attempts also return "Nothing happens." and are already included. Repeated actions with normal environment feedback are not classified as invalid. Rates pool attempts within each task type, and the All group pools all six task types; failed episodes and horizon-limited trajectories remain included. These are descriptive attempt-level rates, not independent episode-level replicates. Within the narrower set of well-formed `take` commands, invalid attempts remain 203/448 for GRPO and 36/279 for WorldPO, with 245 and 243 confirmed acquisitions, respectively.

Tab. 29 summarizes five illustrative pairs with successful WorldPO outcomes, selected to show concrete differences in task progress and completion. Aggregate behavioral comparisons use the complete evaluations described above; these selected examples describe observable decisions within individual trajectories. Throughout this appendix, t is the zero-based action index, so an episode ending at $t = 4$ contains five interaction turns. Task descriptions below are abridged for readability; quoted commands are exact executed actions.

Table 29: **Paired cases across three environments.** Each entry reports task success and the number of interaction turns. Episode IDs identify the original zero-based evaluation slots. The two Search cases belong to 2WikiMultiHopQA.

Environment	ID	Task	WorldPO	Policy-only
ALFWorld	104	Put a bowl on the coffee table	Success / 23	Failure / 50
	84	Examine a bowl with a desk lamp	Success / 5	Success / 10
WebShop	46	Buy a red top in size 3X-large	Success / 5	Success / 6
Search	37052	Compare two film directors' birth dates	Success / 3	Failure / 4
	40289	Compare two film directors' nationalities	Success / 3	Success / 4

I.1 ALFWORLD: TASK PROGRESS AND REPEATED EXAMINATION

The same observed state can lead to earlier task completion. In episode 84, both agents first go to the desk, take the bowl, and examine it twice. At that point, both hold the bowl at the desk where the lamp is available. WorldPO then executes `use desklamp 1`, receives "You turn on the desklamp 1," and succeeds. Policy-only instead examines the bowl, desk, and lamp before examining the bowl twice more and finally using the lamp. The five-turn difference therefore occurs after an identical four-action prefix, not because one agent finds the object sooner.

Finding the relevant object distinguishes the successful trajectory. In episode 104, both agents initially revisit locations and attempt unsuccessful object manipulations. WorldPO eventually visits `diningtable 2` at $t = 19$, where the returned observation lists `bowl 1`. It takes that bowl,

returns to the coffee table, and completes the requested placement at $t = 22$. Policy-only continues attempting to take bowls from drawers, later manipulates a watch, and ends without completing the task after 50 turns. The informative contrast is WorldPO's eventual transition from unsuccessful exploration to an observed object and executable placement sequence, rather than an error-free plan from the outset.

I.2 WebShop: Option Selection and Purchase Completion

An additional attribute click can be unnecessary. In episode 46, both agents open the same top and click color `a6-red` and size `3x-large`. WorldPO purchases immediately after these clicks at $t = 4$. Policy-only clicks the color once more and purchases at $t = 5$. Both receive task score 1, making this an example of a one-turn completion difference after the same product and option commands, rather than a difference in final reward.

I.3 Search: Evidence Acquisition and Answer Completion

Search allocation affects whether an answer is reached. Episode 37052 asks which of two films has the earlier-born director. WorldPO searches the full question, follows with a query about Teja's birth year, and answers *Nene Raju Nene Mantri* at $t = 2$, matching the reference answer. Policy-only first searches for the director of *Invasion 1897* but receives a similarly titled result for *The Invader* (1997); its next query asks about that film's director, Mark Rosman. It then searches for the other film's director and Teja's birth year, ending after four turns without an answer. The trace exposes a retrieval ambiguity and how the remaining search budget is spent.

The same correct answer can require different numbers of searches. Episode 40289 asks whether the directors of *Staying Alive* (2012) and *Shirley Kaye* have the same nationality. WorldPO issues two searches and answers "No" in three turns; policy-only issues three searches and reaches the same answer in four. Some returned items are similarly titled but irrelevant, including a television series in the WorldPO trace. The supported observation is fewer search calls for the same correct answer, not uniformly better retrieval relevance.

I.4 ALFWorld Reasoning Excerpts and Complete Action Sequences

The following paired traces expand ALFWorld episode 84 and retain every executed action, grouping consecutive identical commands with explicit turn-index ranges. Quoted reasoning and observations are verbatim excerpts; descriptions of other observations are identified as summaries.

ALFWorld: Connecting the task goal to an available tool

Task. "examine the bowl with the desklamp." This pair is evaluation episode 84. Both agents start in the same room; the desk, bowl, and lamp needed for the task are encountered along the shared action sequence below. Turn indices t are zero-based. Every executable action is included; a range of repeated turns denotes one execution at each index. Quoted reasoning and observations are verbatim excerpts from the recorded generated responses and returned observations.

Shared actions and observed evidence, $t = 0$–3. At $t = 0$, both execute `go to desk 1`. The environment returns: "You arrive at desk 1. On the desk 1, you see a alarmclock 1, a bowl 1, a cd 2, a cd 1, a cellphone 1, a desklamp 1, a houseplant 1, a keychain 2, a keychain 1, and a pen 1." At $t = 1$, both execute `take bowl 1 from desk 1`, receiving "You pick up the bowl 1 from the desk 1." At both $t = 2$ and $t = 3$, both execute `examine bowl 1`. Each execution returns "This is a normal bowl 1. In it, you see nothing." Thus, before the first action divergence, both agents have acquired the bowl, observed the available lamp, and twice received the same information about the bowl's contents.

WorldPO, $t = 4$: use the task-relevant tool. The generated reasoning states: "The current situation is that I am examining a normal bowl 1, and there is nothing inside. I have already taken steps to pick up the bowl from the desk and examine it. Now, I need to find a way to use the desklamp to examine the bowl further." The extracted action is `use desklamp 1`. The environment returns "You turn on the desklamp 1." and records task success. The complete WorldPO trajectory therefore contains five turns, $t = 0, \ldots, 4$.

GRPO, $t = 4$–9: continue inspecting the bowl and its surroundings. At $t = 4$, the generated reasoning begins "The user has already examined the bowl and found nothing inside." It then lists available alternatives, including using the desk lamp, but the extracted action is again `examine bowl 1`. The returned observation again describes an empty bowl.

At $t = 5$, the reasoning concludes "The agent has already examined the bowl 1, so the next logical step would be to examine the desk 1." The action `examine desk 1` returns the objects remaining on the desk, including `desklamp 1`. At $t = 6$, the reasoning states "The agent needs to examine the desklamp to see if it has any hidden objects or if it is useful." The action `examine desklamp 1` receives "Nothing happens."

At $t = 7$, the generated reasoning says "The agent is now at step 8 and has not examined the bowl." The retained history at this turn contains the two most recent action records, examining the desk and the lamp. GRPO executes `examine bowl 1` at both $t = 7$ and $t = 8$; both executions again return "This is a normal bowl 1. In it, you see nothing." Finally, at $t = 9$, it executes `use desklamp 1`, receives "You turn on the desklamp 1.", and succeeds after ten turns.

Interpretation. The decisive difference occurs after the same four-action prefix and corresponding observations. WorldPO's generated reasoning connects the completed acquisition and examination to using the lamp specified by the task, and its next action completes that task. GRPO instead supplies reasons for further inspection before eventually using the same lamp. This case illustrates a difference in how the explicit reasoning relates the task goal and observed progress to the next executable action, rather than a difference in finding the bowl or accessing the lamp.