

Agentic Reinforcement Learning with Self-Distilled Reward Shaping

Ranxu Zhang*
University of Science and
Technology of China
Hefei, China

Jinghao Lin
Alibaba Group
Hangzhou, China

Guinan Chen
University of Science and
Technology of China
Hefei, China

Xiaozhou Xu
Alibaba Group
Hangzhou, China

Chenshaodong
Alibaba Group
Hangzhou, China

sunzhe
Alibaba Group
Hangzhou, China

Yanyong Zhang
University of Science and
Technology of China
Hefei, China

Chao Wang[†]
University of Science and
Technology of China
Hefei, China

Abstract

Agentic reinforcement learning enables LLM agents to learn through interaction, but sparse trajectory-level rewards reveal success without identifying which intermediate decisions deserve credit. Training-only privileged skills can provide denser supervision by allowing the same frozen policy snapshot to rescore fixed tokens from skill-free trajectories while conditioned on task-matched procedural skills. Existing methods, however, do not jointly calibrate teacher scores across interaction steps, relate teacher confidence to realized returns, and integrate the resulting signal into native reward-to-advantage construction. We introduce Agentic Reinforcement Learning with Self-Distilled Reward Shaping (ADRS), a framework for constructing return-associated token-level credit for multi-turn language agents. ADRS centers and normalizes privileged token scores within each step, modulates them with a return-associated Teacher Value Advantage (TVA) gate based on within-group confidence–return association, and incorporates the gated token signal into native RL credit construction. Together, these components determine what the teacher prefers, when that preference is return-relevant, and how it enters the native reinforcement-learning credit path, while keeping rollouts and inference skill-free. Finally, experiments across three interactive benchmarks show that ADRS consistently improves performance on long-horizon tasks, with gains persisting across RL backbones, reduced-data settings, unseen tasks, and extended training. Our code is available at the following link: <https://github.com/gitrxh/ADRS-arxiv>

CCS Concepts

• Computing methodologies → Reinforcement learning; Artificial intelligence.

Keywords

Agentic Reinforcement Learning, On-Policy Self-Distillation, Large Language Models, Temporal Credit Assignment

*This work was conducted during an internship at Alibaba Group.

[†]Chao Wang is the corresponding author.

1 Introduction

Language-model agents can solve interactive tasks involving evidence search [9], embodied navigation [25], and web interaction [38]. However, agentic reinforcement learning typically provides only terminal outcome rewards, which indicate whether a trajectory succeeds but not which intermediate query or action deserves credit. This sparse feedback makes token-level temporal credit assignment difficult even for critic-free group-relative methods such as GRPO [24] and GiGPO [5]. Natural-language skills can provide denser guidance by describing subgoals, action rules, and common failure patterns [15, 29, 32]. We therefore consider a training-only privileged setting in which the same frozen policy snapshot rescues fixed tokens from skill-free rollouts under task-matched skills; skills remain absent at inference.

This setting connects knowledge distillation [6] with learning using privileged information [28]. Building on on-policy distillation [1], OPSD [40] established a basic recipe in which the same model uses privileged context to supervise trajectories generated without that context. Recent OPSD+RL methods incorporate this supervision in different ways: GRPO+OPSD directly combines the distillation and policy-optimization objectives; Skill-SD [29] extracts dynamic skills from completed trajectories; RLSD [36] adjusts token-level updates using teacher–student differences; and SDAR [15] introduces retrieved procedural skills with confidence-gated auxiliary distillation. Although these methods enrich the source and weighting of privileged supervision, a central question remains: how can privileged token confidence be associated with realized returns and converted into temporal credit through a single native reward-to-advantage-to-policy path?

This question decomposes into three technical challenges as shown in Figure 1: score calibration, reliability estimation, and credit integration. First, raw teacher log-probabilities depend on local context, policy uncertainty, and response composition. Their offsets and scales are therefore not directly comparable across interaction steps. Second, a confident teacher is not necessarily a useful teacher: high token likelihood may reflect linguistic familiarity rather than decisions associated with higher environment

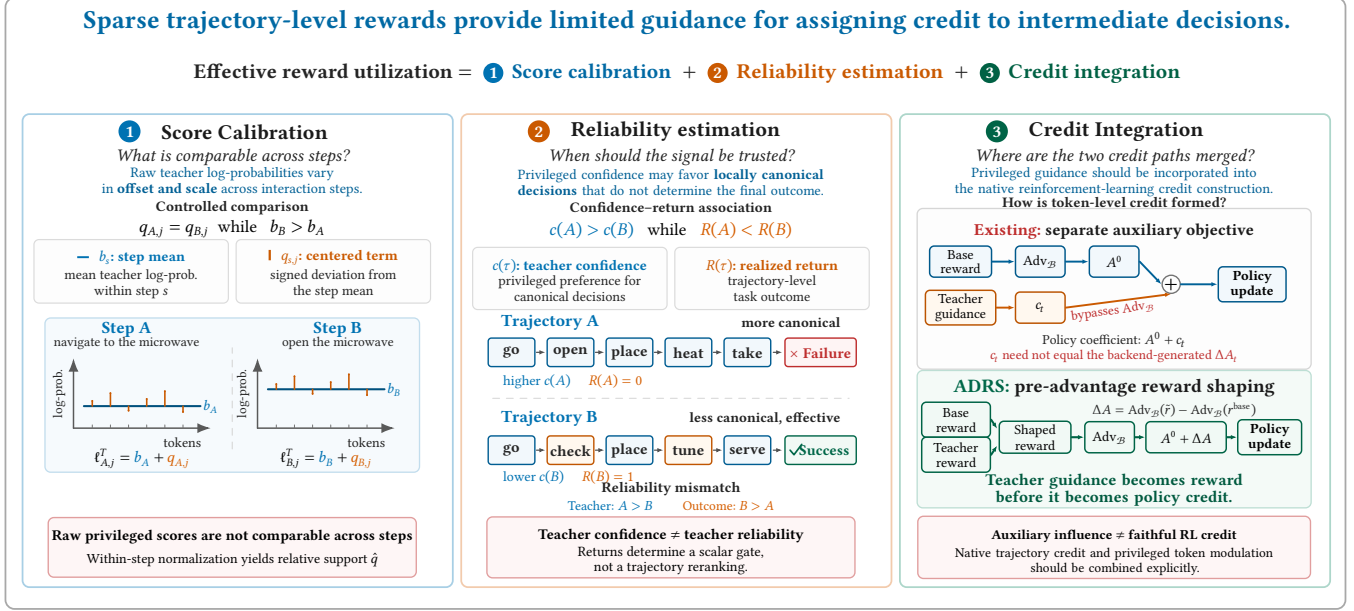


Figure 1: Three challenges for using privileged teacher signals in agentic reinforcement learning. (1) Score calibration: raw teacher scores are not directly comparable across interaction steps. (2) Reliability estimation: teacher confidence may not align with realized returns. (3) Credit integration: privileged signals should be incorporated into native RL credit construction rather than a separate auxiliary objective.

returns. Third, a separate distillation objective places privileged supervision outside the backbone’s reward-to-advantage computation and may misalign token-level updates with the group-relative advantages constructed by GRPO or GiGPO. Addressing these challenges requires a mechanism that calibrates teacher scores across steps, estimates their association with realized returns, and injects the resulting token signal before advantage construction.

To this end, we introduce ADRS, a self-distilled reward-shaping framework that transforms task-matched procedural skills into token-level training rewards while keeping agent rollouts and inference skill-free. ADRS addresses the three challenges through three coupled designs. First, it re-scores student-generated tokens under the same behavior-policy snapshot with privileged skill context, then applies within-step centering and scale normalization to retain relative token preferences while suppressing step-dependent drift. Second, a selected-level return-associated Teacher Value Advantage (TVA) gate modulates the normalized signal using the within-group association between privileged confidence and realized return, rather than assuming uniform teacher reliability. Third, the gated signal is added to the base token rewards before the configured advantage operator, keeping privileged guidance within a single reward-to-advantage-to-policy path rather than a separate auxiliary objective. Together, these components determine what the privileged teacher prefers, when that preference is return-relevant, and how it enters the native reinforcement-learning credit path. Our contributions are:

- We propose ADRS, which converts task-matched procedural skills into return-associated token rewards through within-step score calibration and TVA gating, while keeping rollouts and inference skill-free. For the centered L2/L3 variants, we further establish a stepwise zero-sum structure.

- We formulate the gated signal as pre-advantage reward shaping, integrating privileged guidance into the native reward-to-advantage-to-policy path rather than a separate auxiliary objective. We characterize its compatibility with GRPO/GiGPO-style credit operators and its local first-order connection to sampled-token auxiliary updates.
- Experiments across three interactive benchmarks show that ADRS delivers consistent performance gains on long-horizon tasks across RL backbones, remains effective with reduced training data, generalizes to unseen tasks, and sustains its gains over extended training.

2 Related Work

2.1 Reinforcement Learning for LLMs

Reinforcement learning provides the optimization backbone for training LLMs from environment feedback. PPO [23] uses a clipped policy objective, whereas GRPO [24] removes the critic through within-group outcome normalization. For multi-turn interaction, LOOP [3] develops value-free training for interactive tasks, while RAGEN [31] examines instability and degeneracy during optimization. As interaction horizons grow, localizing outcome feedback becomes increasingly important. GiGPO [5] introduces episode- and step-level relative advantages, while Agent Lightning [16] decomposes trajectories into trainable transitions. These methods improve temporal credit assignment but remain driven by sparse outcomes. ADRS supplements them with privileged token-level guidance before advantage computation.

2.2 Privileged Self-Distillation

Privileged self-distillation combines teacher–student transfer with information available only during training. Knowledge

distillation transfers teacher knowledge to a student [6], while privileged-information learning provides additional training-only supervision [28]. On-policy distillation, DAgger-style learner-distribution supervision, and context distillation extend these ideas to student-generated states and richer contexts [1, 22, 26]. OPSD [40] brings them together by constructing teacher and student views from the same model under different contexts.

Recent methods combine distillation with reward-based learning through SDPO [8] and KDRL [33], or stabilize and selectively apply supervision through TCOd [30], HDPO [4], and TIP [34]. OPCS [39] instead focuses on internalizing enriched context along student-generated trajectories. In language-agent settings, Skill-SD [29] uses trajectory-derived skills, RLSD [36] uses teacher-student differences to modulate token updates, and SDAR [15] uses gated retrieved skills. The differences between SDAR and ADRS are summarized in Table 1. Self-Distilled Policy Gradient [14], Rebellious Student [11], and CRAFT [18] further investigate full-vocabulary supervision, reversed teacher signals, and counterfactual credit from sibling rollouts, respectively. In contrast, ADRS converts calibrated, return-associated teacher scores into token rewards before native advantage construction.

2.3 Reward Shaping and Credit Assignment

Reward shaping supplements sparse feedback with denser signals. Potential-based shaping can preserve optimal policies when based on a fixed Markov potential [19], while RUDDER [2] redistributes delayed returns toward outcome-relevant decisions. Process reward models supervise intermediate reasoning steps [13], but require a separately trained verifier and process-level supervision. Unlike separate reward models or auxiliary distillation objectives, ADRS derives token-level rewards from a privileged policy branch and injects them before advantage construction.

3 Problem Formulation

In multi-turn language-agent reinforcement learning, each task $x_i \sim \mathcal{D}$ induces a partially observed interaction with an environment. Without privileged text, a behavior-policy snapshot π_{θ_b} samples K trajectories $\{\tau_{i,k}\}_{k=1}^K$ for each task. At step s , the history $h_{i,k,s}$ contains the task, observation, and prior actions, and the policy emits $y_{i,k,s} = (y_{i,k,s,1}, \dots, y_{i,k,s,L_{i,k,s}})$ with valid token set $\mathcal{V}_{i,k,s}$. Environment feedback is mapped to base token rewards r^{base} under the same valid-token mask used by teacher scoring, advantage estimation, and the actor loss. For $\mathcal{B} \in \{\text{GRPO}, \text{GiGPO}\}$ [5, 24], let $A^{\mathcal{B}} = \text{Adv}_{\mathcal{B}}(r^{\text{base}})$ denote the native token-aligned advantage, including the backbone’s aggregation, discounting, grouping, normalization, and broadcasting rules.

In each training iteration, π_{θ_b} first performs the ordinary rollout and receives environment feedback. After the trajectories are fixed, a deterministic provider returns task-matched procedural knowledge $\rho(x_i)$, and the same frozen snapshot rescores the realized tokens under $\rho(x_i) \oplus h_{i,k,s}$. This branch changes only the context: it neither samples another trajectory nor introduces a separate or larger teacher, and privileged text is absent from student rollouts and inference. The paired scores, environment returns, and grouping metadata provide training-only evidence for refining credit among the realized token decisions. Our goal is

to use this evidence to assign token-level credit to return-relevant intermediate decisions, thereby improving the skill-free policy π_{θ} while preserving the native GRPO/GiGPO trajectory direction and requiring no privileged input during rollout or inference.

4 Method

ADRS addresses how to convert training-only privileged guidance into return-associated token credit without changing skill-free rollout and inference or replacing the underlying RL backbone. As shown in Figure 2, it operates as a credit-construction layer between trajectory collection and actor optimization. In each training iteration, the behavior policy first samples K trajectories without skill text and receives environment outcomes and base token rewards. Once these trajectories are fixed, the same frozen policy snapshot re-scores the realized tokens under ordinary and privileged contexts. The paired scores and observed outcomes provide the evidence needed to determine which tokens receive stronger privileged support, whether that support is associated with return, and how it should affect the subsequent policy update.

These three questions map directly to three modules. Module 1, *Within-Step Relative Privileged Scoring*, calibrates privileged token scores within each interaction step to obtain comparable relative preferences. Module 2, *Return-Associated TVA Reliability Gating*, uses the raw privileged scores and observed returns to estimate teacher reliability within the selected comparison group. Because Module 2 does not depend on the calibrated output of Module 1, the two modules operate in parallel and meet only in Module 3. Module 3, *Teacher-Guided Per-Token Advantage Modulation*, uses the reliability estimate to modulate the calibrated token signal and combines the resulting token-level redistribution with the environment-defined trajectory direction. The resulting detached advantage is passed to the standard GRPO/GiGPO actor update, after which the updated actor becomes the behavior policy for the next iteration. Privileged text is used only for post-rollout rescaling and remains absent during rollout and inference.

4.1 Within-Step Relative Privileged Scoring

ADRS first considers the comparability of privileged scores across interaction steps. Teacher log-probabilities undergo global shifts and scale changes with context length, interaction position, and response composition. A token with a high raw score in one step can therefore carry less relative support than a token with a lower raw score in another. To obtain a stable token signal, ADRS treats privileged scores as relative evidence inside each realized decision step: it removes the within-step offset, normalizes dispersion, and outputs the signed, scale-controlled score $\hat{q}_{s,j}$. Raw privileged scores are retained for the subsequent return-association estimate.

For each sampled token, the ordinary-context and privileged-context behavior-policy scores are

$$\begin{aligned} \ell_{s,j}^b &= \text{sg}[\log \pi_{\theta_b}(y_{s,j} \mid h_s, y_{s,<j})], \\ \ell_{s,j}^T &= \text{sg}[\log \pi_{\theta_b}(y_{s,j} \mid \rho(x) \oplus h_s, y_{s,<j})]. \end{aligned} \quad (1)$$

The snapshot θ_b generates the batch and remains frozen during both rescaling passes. Thus, ℓ^T measures the support that the privileged context assigns to a token sampled by the student policy; all scores are defined on the same realized student trajectory.

Table 1: Positioning relative to the closest privileged self-teaching methods. “Backward” refers to the policy update.

Method	Skill source	Teacher signal	Injection point	Policy backward
SDAR [15]	Retrieved external SkillBank	Gated teacher-student token gap	Auxiliary distillation loss	RL + auxiliary term
ADRS (ours)	Task-indexed privileged provider	Centered teacher token score + TVA	Native RL credit construction	Single RL objective

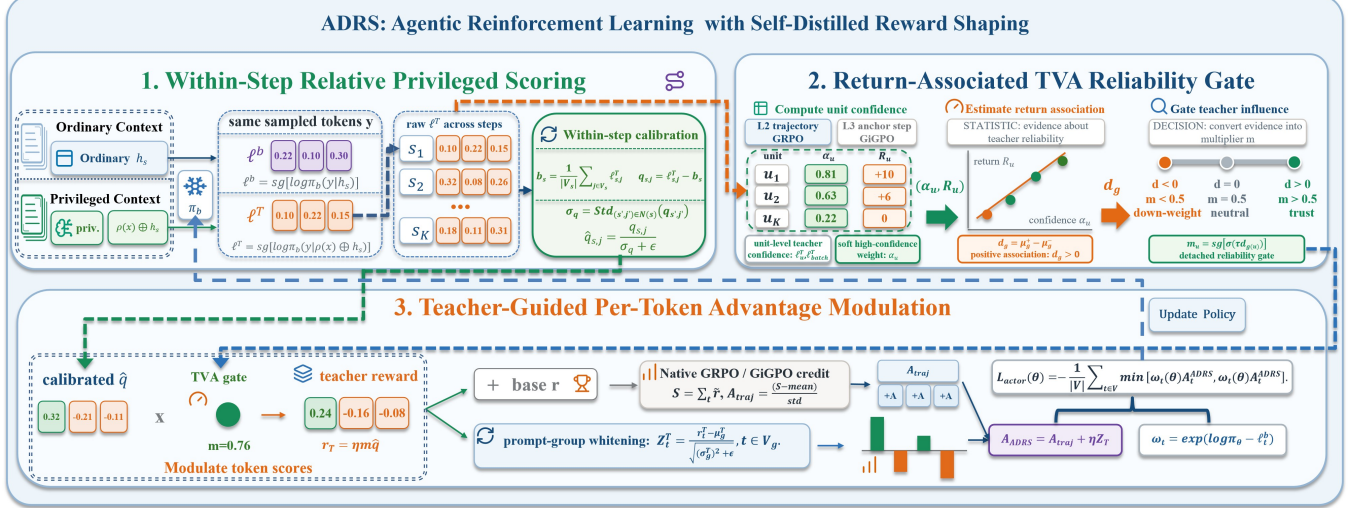


Figure 2: ADRS training flow. A behavior policy samples without privileged text. The same frozen snapshot re-scores those tokens under a task-matched skill. Step-centered scores provide the dense signal, while observed returns calibrate its optional TVA gate. The gated signal is added to the base reward before group-relative backbones such as GRPO and GiGPO estimate advantages; a standard clipped actor update then follows.

The centered and normalized score is

$$b_s = \frac{1}{|\mathcal{V}_s|} \sum_{j \in \mathcal{V}_s} \ell_{s,j}^T, \quad q_{s,j} = \ell_{s,j}^T - b_s, \quad (2)$$

$$\sigma_q = \text{Std}_{(s',j') \in \mathcal{N}(s)}(q_{s',j'}), \quad \hat{q}_{s,j} = \frac{q_{s,j}}{\sigma_q + \epsilon}.$$

Here $\epsilon > 0$ and $\mathcal{N}(s)$ is the chosen normalization scope. The denominator is shared within step s , giving

$$\sum_{j \in \mathcal{V}_s} \hat{q}_{s,j} = 0. \quad (3)$$

Within-step centering removes the overall confidence offset specific to an interaction position, while σ_q makes the strength of teacher guidance robust to changes in score dispersion. A positive $\hat{q}_{s,j}$ indicates that a sampled token receives a privileged score above the step average. ADRS therefore identifies which tokens within the same decision step should receive relatively stronger or weaker credit; the next subsection evaluates the consistency of this relative support with environment return.

4.2 Return-Associated TVA Reliability Gate

ADRS next considers the reliability of privileged confidence. High teacher confidence becomes useful RL guidance when it aligns with observed return in comparable units. To evaluate this alignment, ADRS compares soft teacher confidence with pre-ADRS return inside a selected group and produces a scalar gate for its tokens. This reliability operation controls teacher influence before the final credit-construction operation.

TVA supports two granularities. Completion-level TVA (L2) matches the comparison available in GRPO: trajectories sampled for the same prompt form the group, and each trajectory uses its pre-ADRS base return. Step-level TVA (L3) matches GiGPO’s additional structure: repeated anchor states define comparable interaction steps, and each step uses return-to-go from that state. In this sense, L2 uses prompt-level trajectory information, while L3 uses the anchor-state and step-return information introduced by GiGPO. A training run selects one level.

Let $\mathcal{U}$ denote the units at the selected level and $\mathcal{V}_u$ the valid tokens of unit u . The reliability operation first computes unit confidence and a soft high-confidence weight:

$$\bar{\ell}_u^T = \frac{1}{|\mathcal{V}_u|} \sum_{(s,j) \in \mathcal{V}_u} \ell_{s,j}^T, \quad \bar{\ell}_{\text{batch}}^T = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \bar{\ell}_u^T, \quad \alpha_u = \sigma \left(\frac{\bar{\ell}_u^T - \bar{\ell}_{\text{batch}}^T}{T} \right). \quad (4)$$

It then contrasts returns on the high- and low-confidence sides:

$$\mu_g^+ = \frac{\sum_{u \in \mathcal{G}} \alpha_u R_u}{\sum_{u \in \mathcal{G}} \alpha_u + \epsilon}, \quad \mu_g^- = \frac{\sum_{u \in \mathcal{G}} (1 - \alpha_u) R_u}{\sum_{u \in \mathcal{G}} (1 - \alpha_u) + \epsilon}, \quad d_g = \mu_g^+ - \mu_g^-. \quad (5)$$

After optional positive scale normalization to $\tilde{d}_g$, the detached gate is $m_u = \text{sg}[\sigma(\tau \tilde{d}_{g(u)})]$. The gate is broadcast to all tokens in unit u . Positive return association increases it above 0.5, negative association lowers it, undersized groups use 0.5, and the TVA-off configuration uses $m = 1$.

PROPOSITION 1 (TVA AS RETURN ASSOCIATION). Ignoring the numerical ϵ in Eq. 5, let $\tilde{\alpha}_g = \mathbb{E}_g[\alpha]$. For any non-degenerate group,

$$d_g = \frac{\text{Cov}_g(\alpha, R)}{\tilde{\alpha}_g(1 - \tilde{\alpha}_g)}. \quad (6)$$

Thus, the sign of TVA agrees with the within-group empirical covariance between privileged confidence and return.

Proposition 1 gives the gate its statistical meaning. Since the denominator is positive, d_g increases exactly in the intended direction: units with higher privileged confidence also tend to have higher returns inside the comparison group. This explains why ADRS can use a group-level gate while allowing pairwise teacher errors to be absorbed by the group statistic. The proposition is observational and batch-local; finite ϵ gives the stabilized implementation. The resulting gate m is then combined with the calibrated token score in the final credit-construction operation.

4.3 Teacher-Guided Per-Token Advantage Modulation

ADRS finally considers how the calibrated and reliability-weighted teacher signal should enter policy credit. The within-step score is zero-sum, so a standard trajectory sum preserves outcome ordering while removing the centered signal. ADRS therefore adopts two complementary paths. The trajectory path carries the environment-defined direction of each sampled completion. The token path saves and whitens the privileged signal so that update strength can vary across tokens within the same trajectory.

The teacher reward and shaped token reward are

$$r_{s,j}^T = \eta m_{s,j} \hat{q}_{s,j}, \quad \tilde{r}_{s,j} = r_{s,j}^{\text{base}} + r_{s,j}^T, \quad (7)$$

where $\eta \geq 0$ controls the teacher scale. We keep η in r^T so that the saved tensor matches the injected reward. For positive global η , whitening removes this global scale from Z^T , making the outer η in Eq. 10 the effective token-modulation coefficient.

PROPOSITION 2 (STEPWISE ZERO-SUM REPRESENTATION). *If the normalization denominator and gate are shared within interaction step s , the teacher reward in Eq. 7 satisfies $\sum_{j \in \mathcal{V}_s} r_{s,j}^T = 0$. For any ordering $(j_1, \dots, j_{n_s})$ of the valid tokens, it admits a sampled-chain representation $r_{s,j_r}^T = \Phi_{s,r+1} - \Phi_{s,r}$ with $\Phi_{s,1} = \Phi_{s,n_s+1} = 0$. This identity is an algebraic representation on the sampled step; policy-invariance would require a fixed Markov potential.*

The proposition explains this two-path credit construction. The teacher reward adds zero step-level mass under the stated sharing condition, so trajectory summation keeps the environment ordering. The same zero-sum identity motivates a token-preserving path for training the privileged structure. ADRS therefore sends $\tilde{r}$ through the trajectory path and saves r^T for token modulation.

For GRPO, the trajectory path computes

$$S_{i,k} = \sum_{t \in \mathcal{V}_{i,k}} \tilde{r}_{i,k,t}, \quad \mu_i^S = \text{Mean}_{k=1}^K (S_{i,k}), \quad (8)$$

$$(\sigma_i^S)^2 = \text{Var}_{k=1}^K (S_{i,k}), \quad A_{i,k}^{\text{traj}} = \frac{S_{i,k} - \mu_i^S}{\sqrt{(\sigma_i^S)^2 + \epsilon}}.$$

Because the step-shared teacher reward sums to zero, $S_{i,k}$ equals the base trajectory score in this configuration; A^{traj} carries the trajectory-level outcome direction.

The token path whitens the saved teacher reward within prompt group g :

$$\mu_g^T = \text{Mean}_{v \in \mathcal{V}_g} (r_v^T), \quad (\sigma_g^T)^2 = \text{Var}_{v \in \mathcal{V}_g} (r_v^T),$$

$$Z_t^T = \frac{r_t^T - \mu_g^T}{\sqrt{(\sigma_g^T)^2 + \epsilon}}, \quad t \in \mathcal{V}_g. \quad (9)$$

The final token-level advantage is

$$A_{i,k,t}^{\text{ADRS}} = A_{i,k}^{\text{traj}} + \eta Z_{i,k,t}^T. \quad (10)$$

The first term sets trajectory direction; the second redistributes update strength along it. Since Z^T is standardized, η directly controls token modulation and recovers the native algorithm at $\eta = 0$. For GiGPO, ADRS adds Z^T to the episode-level advantage while retaining the native step-level branch.

For flattened valid token t with ordinary context c_t , define

$$\omega_t(\theta) = \exp(\log \pi_\theta(y_t | c_t) - \ell_t^\theta), \quad \bar{\omega}_t(\theta) = \text{clip}(\omega_t(\theta), 1 - \epsilon_{\text{lo}}, 1 + \epsilon_{\text{hi}}). \quad (11)$$

With detached A^{ADRS} , the actor uses

$$\mathcal{L}_{\text{actor}}(\theta) = -\frac{1}{|\mathcal{V}|} \sum_{t \in \mathcal{V}} \min[\omega_t(\theta) A_t^{\text{ADRS}}, \bar{\omega}_t(\theta) A_t^{\text{ADRS}}]. \quad (12)$$

Thus, privileged information affects learning through the ordinary policy ratio after being converted into detached credit.

PROPOSITION 3 (TRAJECTORY PRESERVATION AND LOCAL TOKEN--GRADIENT EQUIVALENCE). *Under a step-shared gate, Eq. 7 leaves the GRPO trajectory score, and hence A^{traj} , unchanged. If the teacher signal is non-constant within the group, Eq. 10 nevertheless induces nonzero token-level credit. Define $A_t^0 = A_{i,k}^{\text{traj}}$ and $\Delta A_t = \eta Z_t^T$. At the behavior-policy point with clipping inactive,*

$$\nabla_\theta J_{\text{ADRS}}|_{\theta_b} = \frac{1}{|\mathcal{V}|} \sum_{t \in \mathcal{V}} (A_t^0 + \Delta A_t) \nabla_\theta \log \pi_\theta(y_t | c_t). \quad (13)$$

For a detached sampled-token auxiliary coefficient a_t , the corresponding local update is

$$\nabla_\theta J_{\text{aux}}|_{\theta_b} = \frac{1}{|\mathcal{V}|} \sum_{t \in \mathcal{V}} (A_t^0 + a_t) \nabla_\theta \log \pi_\theta(y_t | c_t). \quad (14)$$

Setting $a_t = \Delta A_t = \eta Z_t^T$ makes the two first-order policy gradients identical at this evaluation point.

Proposition 3 clarifies the optimization role of ADRS. The trajectory path carries the return-defined decision, and the token path contributes a local score-weighted update equivalent to a detached token auxiliary coefficient at the behavior-policy point. The statement is local to the sampled batch and unclipped branch; broader objective equivalences require separate assumptions. ADRS then passes A^{ADRS} to GRPO/GiGPO, closing the iteration and preparing the next behavior snapshot.

5 Experiments

5.1 Experimental Settings

Benchmarks. We evaluate ADRS across three complementary language-agent interactions. ALFWorld [25] casts household tasks as text-based embodied control, requiring agents to satisfy ordered preconditions over extended action sequences. WebShop [38] evaluates web navigation, where an agent searches, inspects product attributes, retains user constraints, and completes a purchase. Search-based QA follows Search-R1 [9] and tests evidence gathering over seven question-answering datasets using only four tool-use turns. The combination separates long-horizon procedural control from shorter information-seeking

interaction, allowing us to test where privileged token credit is most useful. Appendix A.1 provides dataset-level descriptions, split usage, metrics, and interaction limits.

Baselines. We organize comparisons by supervision source and use. Vanilla and Skill-Prompt test the instruction-tuned policy without or with inference-time skills. GRPO [24], Skill-GRPO, and GiGPO [5] represent outcome-driven reinforcement learning, with GiGPO supplying a step-aware credit backbone. OPSD [40], GRPO+OPSD, Skill-SD [29], RLSD [36], and SDAR [15] provide structured self-distillation or skill-conditioned supervision. These groups distinguish gains from outcome optimization, skill access, auxiliary token supervision, and native RL credit construction. Unless marked by an asterisk, evaluation uses no skill context; ADRS never uses privileged skills during rollout or inference. Appendix A.3 provides details.

Models, metrics, and protocol. The common 150-step comparison covers Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct [21], and Qwen3-1.7B-Instruct [35]. We report ALFWorld task-family and overall success, Search per-dataset exact match and macro-average, and WebShop normalized score and exact success. We extend representative runs to 300 steps, separate GRPO and GiGPO backbones, inspect action/object token gaps, vary available training data, evaluate unseen transfer, and sweep maximum interaction horizon against SDAR. All experiments follow the SDAR benchmark and evaluation setup [15], using multiple NVIDIA H100 GPUs. Appendix A provides additional implementation details.

5.2 Overall Performance

Table 2 reports the common 150-step comparison. Three findings organize the results.

ADRS establishes new state-of-the-art performance across all three benchmarks with Qwen2.5-3B. With Qwen2.5-3B, ADRS achieves 94.5% overall success on ALFWorld, a 45.0% macro-average on Search-based QA, and a score/success pair of 87.5/76.6 on WebShop. Compared with the strongest baselines, these results yield improvements of 10.1 percentage points on ALFWorld, 0.4 points on Search-based QA, and 2.5/8.6 points on the two WebShop metrics, respectively, establishing new state-of-the-art results across all three benchmarks.

ADRS delivers consistent ALFWorld gains across model families and scales. ADRS attains the highest ALFWorld overall success in all three model blocks: 94.5% with Qwen2.5-3B, 96.1% with Qwen2.5-7B, and 62.5% with Qwen3-1.7B. These results exceed the strongest corresponding baselines by 10.1, 7.8, and 8.6 percentage points, respectively. With Qwen3-1.7B, ADRS also achieves the highest Search-based QA macro-average of 42.9% and the highest WebShop success rate of 65.6%, improving over the strongest baselines by 0.7 and 7.0 points. These results demonstrate that the effectiveness of ADRS extends across different model families and parameter scales.

The performance gains do not rely on privileged information at evaluation time. Skill-Prompt and the starred Skill-GRPO variant provide skills directly to the policy during evaluation, whereas ADRS uses only the ordinary interaction history

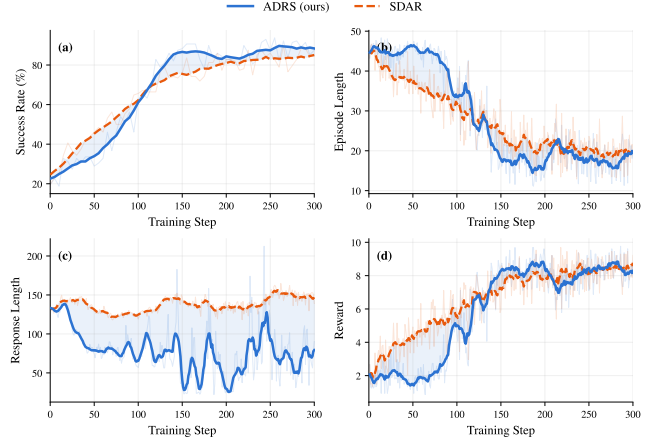


Figure 3: ALFWorld training dynamics through 300 steps. Curves compare SDAR and ADRS+TVA (GiGPO).

during rollout and inference. The leading results of ADRS therefore do not depend on additional test-time skill context. Instead, ADRS converts training-only privileged guidance into a policy that performs effectively under skill-free interaction.

Appendix Figures 9–10 shows the training success rate, episode-length, and response-length trajectories. The training curves exhibit consistent upward trends across all configurations.

5.3 Long-Horizon Training

The 150-step table establishes a common-budget comparison but does not show whether the gains persist during later optimization. We therefore continue representative runs to 300 steps and report the best checkpoint, final checkpoint, and mean over the last three evaluations in Table 3. These statistics distinguish an isolated peak from sustained late-training performance.

ADRS sustains strong late-training performance on the two procedural environments. On ALFWorld, the ADRS-global+entropy/clip run reaches a 97.7% peak and finishes at 96.1%, 9.4 points above the corresponding GRPO final checkpoint. This run maintains a high task success rate. Within the GiGPO sequence, ADRS+TVA improves the final/last-three values from 91.4/93.0% to 94.5/94.3%. On WebShop, an ADRS run reaches an 84.4% peak and final success rate, 5.5 points above the best GRPO checkpoint; the no-TVA run records the strongest last-three average at 80.2%.

Figure 3 visualizes the 300-step ALFWorld trajectories. Appendix Table 9 reports the complete named-run matrix.

5.4 Ablations on GRPO and GiGPO Backbones

Figure 4 shows the WebShop training dynamics under the 150-step budget, comparing GRPO with the corresponding ADRS variants. The ADRS variants remain above GRPO over most of training, and the strongest configuration reaches higher final performance at step 150. This complements the endpoint comparison in the main table, showing that the WebShop gain is visible across the training curve rather than only at a single checkpoint.

Table 2: Performance Comparison on the representative long-horizon benchmarks (ALFWorld, Search-based QA, and WebShop). We report the success rate (%) on ALFWorld, accuracy on search-based QA, and task-completion score/success rate on WebShop. An asterisk (*) denotes validation with skills. The best, second-best, and third-best results are highlighted.

Method	ALFWorld							Search-based QA							WebShop		
	Pick	Look	Clean	Heat	Cool	Pick2	Avg	NQ	Triv	Pop	Hotp	2Wk	MuS	Bam	Avg	Score	Succ.
Qwen2.5-3B-Instruct																	
Vanilla	44.4	11.1	6.2	15.4	28.6	12.5	21.9	24.6	48.1	31.0	26.3	25.3	7.2	59.7	31.7	6.7	0.8
Skill-Prompt*	51.7	66.7	48.4	0.0	4.3	10.0	28.9	23.7	46.2	30.6	24.4	22.1	7.5	12.5	23.9	0.2	0.8
OPSD	48.8	41.7	16.7	0.0	15.8	16.7	28.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	11.3	3.1
GRPO	91.2	62.5	96.2	61.9	65.0	47.4	75.0	39.3	60.6	41.1	37.4	34.6	15.4	26.4	36.4	79.8	63.3
Skill-GRPO	88.9	71.4	58.8	70.6	40.7	29.2	60.2	43.5	58.8	43.0	36.8	32.2	11.7	12.5	34.1	77.3	60.9
Skill-GRPO*	94.3	57.1	100.0	66.7	73.1	57.1	80.5	44.3	59.6	44.3	39.0	36.1	14.5	14.9	36.1	76.3	66.4
GRPO+OPSD	100.0	82.4	85.7	75.0	70.0	60.0	81.2	44.9	61.2	45.2	40.4	38.5	16.0	66.1	44.6	77.8	66.4
Skill-SD	88.2	50.0	96.2	52.4	65.0	57.9	73.4	44.4	60.4	44.0	39.5	40.4	15.4	64.9	44.1	75.9	64.0
RLSD	87.9	75.0	90.9	75.0	73.1	68.4	79.7	41.5	58.6	42.3	40.4	40.2	16.8	66.9	43.8	84.4	66.4
SDAR	97.1	62.5	100.0	61.9	75.0	84.2	84.4	44.8	58.1	44.3	38.6	36.2	15.7	66.1	43.4	85.0	68.0
ADRS (ours)	94.1	75.0	100.0	100.0	90.0	94.7	94.5	44.7	61.4	45.8	39.9	37.3	14.4	64.9	45.0	87.5	76.6
Qwen2.5-7B-Instruct																	
Vanilla	36.1	22.2	3.1	0.0	0.0	0.0	12.5	25.2	50.8	29.5	29.0	29.0	10.4	63.7	33.9	5.9	1.6
Skill-Prompt*	51.7	50.0	32.3	5.3	4.3	0.0	23.4	30.9	52.1	32.7	32.7	27.9	12.7	66.1	36.4	1.7	0.8
OPSD	50.0	60.0	22.7	21.4	17.6	9.5	32.8	8.8	8.6	17.5	2.5	4.2	0.5	1.2	6.2	4.5	2.3
GRPO	91.2	87.5	96.2	81.0	65.0	57.9	81.2	45.1	63.7	44.0	43.6	43.2	16.8	37.6	42.0	80.9	72.6
Skill-GRPO	88.5	66.7	65.2	61.1	57.7	73.1	69.5	45.2	63.7	45.7	43.1	43.3	19.6	21.4	40.3	80.4	71.9
Skill-GRPO*	100.0	83.3	96.4	83.3	75.0	78.9	88.3	44.8	63.0	45.1	43.7	43.7	20.5	71.4	47.5	87.0	81.2
GRPO+OPSD	91.4	61.5	100.0	87.5	76.5	52.2	80.4	47.3	64.5	46.9	43.8	39.3	18.0	69.4	47.0	86.8	76.5
Skill-SD	93.9	93.8	90.9	100.0	69.2	68.4	85.1	47.1	64.5	47.8	44.2	42.1	20.2	69.0	47.8	86.1	76.5
RLSD	100.0	87.5	92.3	58.8	80.0	65.2	82.0	46.8	63.0	44.4	45.5	48.9	21.5	73.0	49.0	87.4	77.3
SDAR	94.7	75.0	100.0	86.7	68.2	78.9	85.9	46.3	63.5	48.2	43.8	48.4	19.6	73.0	49.0	89.4	82.8
ADRS (ours)	100.0	87.5	100.0	95.2	90.0	94.7	96.1	47.5	64.2	46.5	43.6	43.8	18.7	69.4	48.2	88.0	79.7
Qwen3-1.7B																	
Vanilla	25.0	22.2	3.1	0.0	21.4	4.2	12.5	29.4	46.9	37.0	23.5	19.6	6.4	10.5	24.8	46.5	4.7
Skill-Prompt*	10.3	50.0	16.1	0.0	0.0	5.0	9.4	29.4	46.5	36.2	22.9	20.8	4.3	10.1	24.3	23.0	2.3
OPSD	26.3	33.3	9.1	0.0	4.5	5.3	14.1	4.2	8.3	4.6	6.6	15.3	0.7	1.2	5.8	47.4	9.3
GRPO	71.1	41.7	36.4	40.0	31.8	31.6	46.1	40.0	58.9	43.5	35.4	30.3	12.0	65.7	40.8	67.3	38.3
Skill-GRPO	27.6	54.5	22.7	27.3	0.0	19.2	21.1	39.2	58.6	43.9	35.2	28.2	11.5	66.1	40.4	73.4	46.1
Skill-GRPO*	31.4	42.9	51.9	8.3	11.5	7.1	28.1	38.0	58.4	43.9	36.3	29.0	12.5	66.9	40.7	80.4	50.0
GRPO+OPSD	38.2	50.0	30.8	28.6	30.0	21.1	32.0	40.7	58.9	45.0	37.0	34.6	13.3	65.7	42.2	70.7	38.3
Skill-SD	52.9	37.5	69.2	42.9	60.0	36.8	52.3	39.1	57.5	45.4	34.8	34.1	10.7	64.1	40.8	81.8	53.9
RLSD	50.0	37.5	61.5	19.0	50.0	21.1	42.2	38.6	57.3	43.0	34.5	34.1	11.5	65.3	40.6	74.0	50.8
SDAR	73.5	25.0	76.9	33.3	40.0	36.8	53.9	39.7	58.9	45.3	35.9	35.5	12.6	65.3	41.9	76.8	58.6
ADRS (ours)	86.8	50.0	59.1	73.3	63.6	15.8	62.5	42.7	59.4	47.4	35.7	32.2	12.5	65.7	42.9	77.7	65.6

Table 3: Representative 300-step convergence results. “Best” is the highest observed checkpoint.

Env.	Configuration	Best	Final	Last-3
ALF.	GRPO	92.2	86.7	86.2
	GiGPO	96.9	91.4	93.0
	ADRS-global (GRPO)	97.7	96.1	96.1
	ADRS+TVA (GiGPO)	96.9	94.5	94.3
Web.	GRPO	78.9	78.9	75.3
	ADRS (no TVA)	82.8	82.8	80.2
	ADRS-per-seq	84.4	84.4	77.9
	ADRS-per-seq+GateNorm	84.4	82.0	78.9

Figure 5 compares GiGPO, ADRS without TVA, and ADRS+TVA on ALFWorld, testing whether ADRS still brings gains when the

backbone already provides step-aware credit. The WebShop trajectories show that ADRS improves a GRPO backbone under the preferred WebShop normalization during optimization. Appendix Tables 8 and 7 report the matched normalization comparison and teacher-scale sensitivity results.

ADRS remains beneficial on the GiGPO backbone. At step 150, centered ADRS improves GiGPO from 82.8% to 87.5%, and ADRS+TVA further reaches 89.8%. Their corresponding best checkpoints are 86.7%, 90.6%, and 93.0%. Across the 300-step runs, TVA increases the peak by 0.8 points and the last-three average by 2.4 points. This comparison supports the complete GiGPO-based system. Appendix Table 10 preserves the fully named matrix.

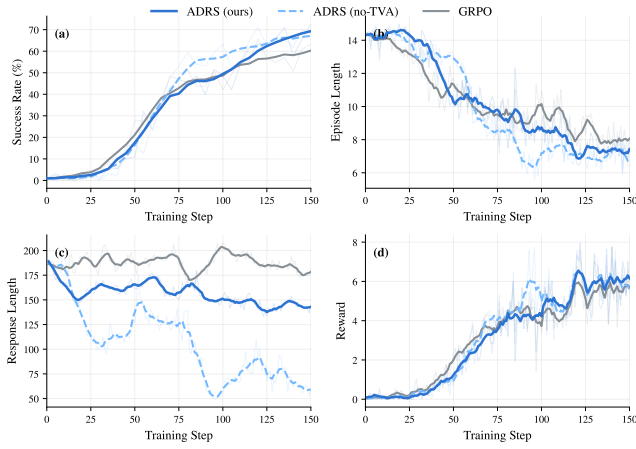


Figure 4: WebShop ablation dynamics through 150 steps. Curves compare GRPO with the ADRS variants.

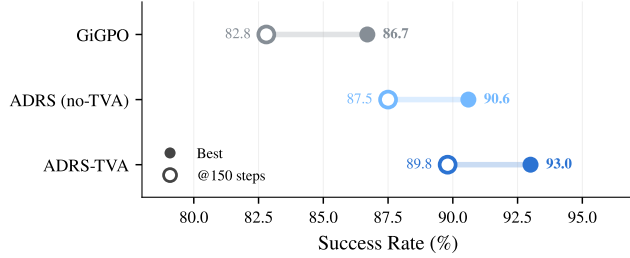


Figure 5: GiGPO-based ALFWorld ablation on Qwen2.5-3B. The chart compares GiGPO, ADRS without TVA, and ADRS with TVA at step 150 and at their best observed checkpoints.

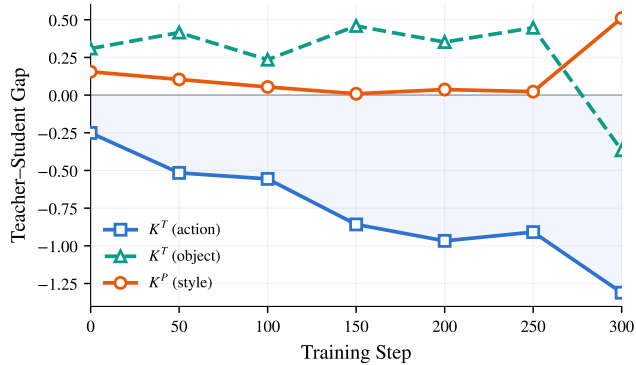


Figure 6: Action-object diagnostic over training on four fixed successful ALFWorld trajectories. K^T groups task-bearing action and object tokens, whereas K^P denotes the residual style-token group. Values are signed teacher-student log-probability gaps.

5.5 Action-Object Mechanism Diagnostic

To examine how token-level advantage modulation supports long-horizon performance, we probe four fixed successful ALFWorld trajectories: heat, pick-and-place, cool, and look-at-object-in-light. These trajectories contain 32 action-token, 80 object-token, and

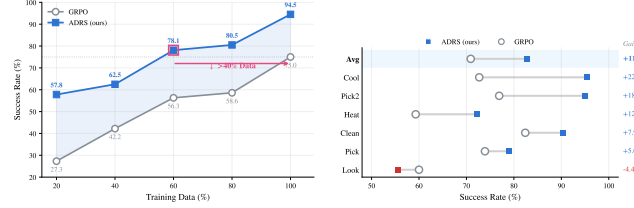


Figure 7: Data efficiency and evaluation on the supplied ALFWorld unseen split. Left: ADRS trained on five fractions of the available data, with full-data GRPO shown as a reference. Right: category-level success on the unseen evaluation.

470 style/other-token observations. Figure 6 groups the signed teacher-student gaps into K^T action/object terms and a residual K^P style term, revealing how privileged rescaling separates task-bearing tokens from generic style tokens over training.

At step 150, the action gap reaches -0.8582 and the object gap remains positive at 0.4591 , while the style gap contracts to 0.0085 , yielding an action-to-style magnitude ratio of 95.3 . This separation shows that the token-level ADRS signal increasingly concentrates update contrast on tokens that determine executable actions and object choices, rather than spreading credit uniformly across the generated response. The pattern explains why token-level advantage modulation improves long-horizon interaction: it strengthens learning on the local decisions that carry delayed task consequences. After step 250, task success approaches saturation and the policy changes more slowly; the shift in the object term and the smaller final ratio coincide with this later training stage.

5.6 Data Efficiency and Unseen Generalization

We next evaluate whether ADRS can extract useful token-level credit from smaller training sets and whether the learned behavior transfers to the supplied unseen ALFWorld split.

ADRS surpasses full-data GRPO after using 60% of the training data. Figure 7 (left) evaluates ADRS with 20%, 40%, 60%, 80%, and 100% of the available ALFWorld training data. ADRS obtains 57.8%, 62.5%, 78.1%, 80.5%, and 94.5% success, respectively. The 60%, 80%, and 100% settings exceed the 75.0% full-data GRPO reference, while the 20% and 40% settings remain below it. This trend shows that ADRS benefits from additional training data but can exceed the full-data outcome-only baseline before using the entire training set.

ADRS transfers to most categories in the supplied unseen split. Figure 7 (right) compares ADRS and GRPO on the unseen ALFWorld split. ADRS improves the average success rate by 11.9 points. The category-level differences are +22.8 on Cool, +18.1 on Pick2, +12.9 on Heat, +7.9 on Clean, +5.0 on Pick, and -4.4 on Look. Thus, ADRS improves five of the six reported categories, with the largest gains on state-changing and multi-object tasks.

5.7 Effect of Maximum Interaction Steps

Figure 8 compares ADRS and SDAR with maximum episode lengths of 25, 35, and 50 steps. ADRS consistently outperforms SDAR, achieving 85.2%, 85.2%, and 94.5%, respectively, compared with 78.9%, 75.0%, and 84.4% for SDAR. As the interaction budget

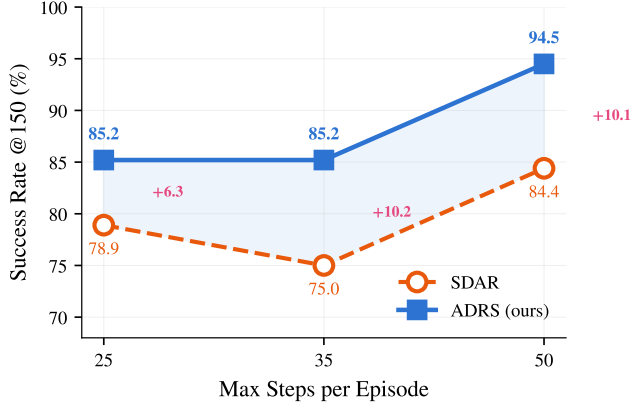


Figure 8: Effect of maximum interaction steps on ALFWorld with Qwen2.5-3B at training step 150.

increases, ADRS maintains its advantage and reaches its strongest result at the 50-step horizon. This result indicates that ADRS can use a longer interaction horizon to improve long-horizon task performance while preserving reliable token-level credit assignment across interaction steps.

6 Conclusion

We presented ADRS, a self-distilled reward-shaping framework for sparse temporal credit in multi-turn language-agent reinforcement learning. ADRS transforms task-matched procedural guidance into token rewards through within-step calibration, return-associated TVA modulation, and pre-advantage integration. These components determine what the privileged teacher prefers, when that preference is return-relevant, and how it enters policy optimization, while keeping rollouts and inference skill-free. Experiments across three interactive benchmarks and multiple training settings demonstrate consistent improvements, particularly on long-horizon tasks. Taken together, these results indicate that training-only procedural knowledge can be converted into return-associated token-level credit and incorporated directly into the native reward-to-advantage construction.

References

- [1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, Vol. 2024. 21246–21263.
- [2] Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. 2019. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems* 32 (2019).
- [3] Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. 2025. Reinforcement Learning for Long-Horizon Interactive LLM Agents. *arXiv preprint arXiv:2502.01600* (2025).
- [4] Ken Ding. 2026. HDPO: Hybrid Distillation Policy Optimization via Privileged Self-Distillation. *arXiv preprint arXiv:2603.23871* (2026).
- [5] Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2026. Group-in-group policy optimization for LLM agent training. *Advances in Neural Information Processing Systems* 38 (2026), 46375–46408.
- [6] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).
- [7] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*. 6609–6625.
- [8] Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. 2026. Reinforcement Learning via Self-Distillation. *arXiv preprint arXiv:2601.20802* (2026).
- [9] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-R1: Training LLMs to Reason and Leverage Search Engines with Reinforcement Learning. *arXiv preprint arXiv:2503.09516* (2025).
- [10] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1601–1611.
- [11] Jeonghye Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. 2026. Rebellious Student: Reversing Teacher Signals for Reasoning Exploration with Self-Distilled RLVR. *arXiv preprint arXiv:2605.10781* (2026).
- [12] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics* 7 (2019), 453–466.
- [13] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let’s verify step by step. In *International Conference on Learning Representations*, Vol. 2024. 39578–39601.
- [14] Yifeng Liu, Shiyuan Zhang, Yifan Zhang, and Quanquan Gu. 2026. Self-Distilled Policy Gradient. *arXiv preprint arXiv:2606.04036* (2026).
- [15] Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, et al. 2026. Self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2605.15155* (2026).
- [16] Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K Qiu, and Yuqing Yang. 2025. Agent Lightning: Train Any AI Agents with Reinforcement Learning. *arXiv preprint arXiv:2508.03680* (2025).
- [17] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hananeh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*. 9802–9822.
- [18] Zibin Meng and Kani Chen. 2026. CRAFT: Counterfactual Credit Assignment from Free Sibling Rollouts for Self-Distilled Agentic Reinforcement Learning. *arXiv preprint arXiv:2606.29476* (2026).
- [19] Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, Vol. 99. Citeseer, 278–287.
- [20] Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2023. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*. 5687–5711.
- [21] Qwen Team. 2025. Qwen2.5 Technical Report. arXiv:2412.15115 [cs.CL] <https://arxiv.org/abs/2412.15115>
- [22] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 627–635.
- [23] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).
- [24] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* (2024).
- [25] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. ALFWorld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768* (2020).
- [26] Charlie Snell, Dan Klein, and Ruiqi Zhong. 2022. Learning by distilling context. *arXiv preprint arXiv:2209.15189* (2022).
- [27] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop Questions via Single-Hop Question Composition. *Transactions of the Association for Computational Linguistics* 10 (2022), 539–554.
- [28] Vladimir Vapnik and Akshay Vashist. 2009. A new learning paradigm: Learning using privileged information. *Neural networks* 22, 5-6 (2009), 544–557.
- [29] Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, et al. 2026. Skill-SD: Skill-Conditioned Self-Distillation for Multi-Turn LLM Agents. *arXiv preprint arXiv:2604.10674* (2026).
- [30] Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. 2026. TCO: Exploring Temporal Curriculum in On-Policy Distillation for Multi-Turn Autonomous Agents. *arXiv preprint arXiv:2604.24005* (2026).
- [31] Zihan Wang, Kangrui Wang, Qingen Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. 2025. RAGEN: Understanding Self-Evolution in LLM Agents via Multi-Turn Reinforcement Learning. *arXiv preprint arXiv:2504.20073* (2025).
- [32] Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. 2026. SkillRL: Evolving Agents via Recursive Skill-Augmented Reinforcement Learning. *arXiv preprint arXiv:2602.08234* (2026).
- [33] Hongling Xu, Qi Zhu, Heyuan Deng, Jinpeng Li, Lu Hou, Yasheng Wang, Lifeng Shang, Ruifeng Xu, and Fei Mi. 2025. KDRL: Post-Training Reasoning LLMs via Unified Knowledge Distillation and Reinforcement Learning. *arXiv preprint arXiv:2506.02208* (2025).
- [34] Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. 2026. TIP: Token Importance in On-Policy Distillation. *arXiv preprint arXiv:2604.14084* (2026).
- [35] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025).
- [36] Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. 2026. Self-Distilled RLVR. *arXiv preprint arXiv:2604.03128* (2026).
- [37] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*. 2369–2380.
- [38] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. WebShop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems* 35 (2022), 20744–20757.
- [39] Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2026. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275* (2026).
- [40] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. 2026. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. *arXiv preprint arXiv:2601.18734* (2026).

A Experimental and Implementation Details

This section expands the compact setup in Section 5.1. It describes the benchmark protocols and baseline groups, then documents the skill provider, named configurations, and optimization settings used in our experiments.

A.1 Benchmarks

Our evaluation covers embodied household control, interactive web navigation, and search-augmented question answering. Table 4 summarizes the evaluation sets, primary metrics, and maximum interaction horizons.

ALFWorld. ALFWorld [25] presents household tasks through a text interface derived from embodied environments. Given an instruction and textual observations, the agent must select admissible actions while satisfying object-state and receptacle preconditions. We report success over 128 validation tasks spanning Pick, Look, Clean, Heat, Cool, and Pick2. The 50-turn limit makes ALFWorld the longest-horizon benchmark in the primary suite and exposes errors in ordering, state tracking, and recovery from invalid actions.

WebShop. WebShop [38] requires an agent to search a simulated product catalog, inspect candidate attributes, preserve constraints from a natural-language request, and complete a purchase. Each checkpoint evaluates 128 tasks sampled from goals 0–499. The environment reports a normalized task score for partial constraint satisfaction and a binary success measure for exact completion. Its 15-turn interaction budget is shorter than ALFWorld but still requires multi-stage search, comparison, and selection.

Search-based QA. Following Search-R1 [9], the agent issues search queries, reads retrieved evidence, and returns an answer within four tool-use turns. The aggregate covers Natural Questions [12], TriviaQA [10], PopQA [17], HotpotQA [37], 2WikiMultiHopQA [7], MuSiQue [27], and Bamboogle [20]. These datasets mix direct retrieval and multi-hop evidence synthesis. The primary runs evaluate the full seven-dataset 51K collection every 75 optimizer steps, whereas the η sweep uses a 504-example development subset every 15 steps.

A.2 Metrics and Reward Construction

For N evaluation tasks and seven Search datasets $\mathcal{D}$, the reported aggregates are

$$\begin{aligned} \text{Succ}_{\text{ALF}} &= \frac{100}{N} \sum_{n=1}^N \mathbf{1}[\text{task } n \text{ succeeds}], \\ \text{EM}_{\text{Search}} &= \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \frac{100}{N_d} \sum_{n=1}^{N_d} \mathbf{1}[\hat{a}_{d,n} = a_{d,n}], \\ \text{Score}_{\text{WS}} &= \frac{100}{N} \sum_{n=1}^N \text{taskscore}_n, \\ \text{Succ}_{\text{WS}} &= \frac{100}{N} \sum_{n=1}^N \mathbf{1}[\text{taskscore}_n = 1]. \end{aligned}$$

ALFWorld also reports success by task family, and Search reports the exact match for each constituent dataset before taking their unweighted macro-average. WebShop score and success are complementary: the former preserves partial task completion, whereas the latter requires every purchase constraint to be satisfied. All evaluations use temperature-0.4 sampling.

The episode reward manager places accumulated outcome reward on the final valid response token before advantage estimation. ALFWorld uses $10 \mathbf{1}[\text{won}]$, WebShop uses $10 \mathbf{1}[\text{done} \wedge \text{taskscore} = 1]$, and Search uses $\mathbf{1}[\text{exact match}]$. Invalid-action penalty coefficients are 0.1 for ALFWorld/WebShop and 0.01 for Search; reference-policy coefficients are 0.01, 0.01, and 0.001, respectively. These terms form r^{base} before Eq. 7.

A.3 Baselines

The baselines differ in whether they optimize only terminal outcomes, expose natural-language skills to the acting policy, or use a privileged teacher to supply token-level supervision.

Prompting and outcome-only reinforcement learning. Vanilla evaluates the original instruction-tuned backbone without post-training. Skill-Prompt uses the same frozen policy but appends a task-relevant skill during evaluation, isolating the benefit of direct inference-time guidance. GRPO [24] samples multiple trajectories per task and derives a group-relative advantage from scalar outcomes. Skill-GRPO retains the GRPO objective while supplying skills during training, and the starred variant also supplies them during evaluation. GiGPO [5] augments outcome-driven learning with step-aware group credit and serves as a stronger backbone in the 150-step component study.

Privileged self-distillation. OPSD [40] uses ordinary and privileged contexts to obtain student and teacher predictions from the same model. GRPO+OPSD combines its auxiliary token objective with outcome-based RL. Skill-SD [29] supplies trajectory-derived skills to the teacher branch, while RLSD [36] uses teacher–student differences to modulate token updates. SDAR [15] introduces retrieved procedural skills and a confidence-gated auxiliary distillation loss. These methods provide the closest comparison for determining whether privileged guidance is more effective when injected into the native reward-to-advantage path.

Comparison setup. All unstarred methods are evaluated without privileged skills. We follow the benchmark, training, and evaluation setup of SDAR [15] and use the same interaction protocols across the compared model configurations.

A.4 Skill Provider and Memory Construction

The provider contains 37 general skills and 90 task-specific skills. All 619 source memories come from the designated training splits: 223 for ALFWorld (113 successful and 110 failed), 196 for Search (132/64), and 200 for WebShop (120/80). The skill configuration follows the official SDAR setup [15], which adopts the SkillBank introduced by SkillRL [32]. The primary runs use compact skill documents rather than appending individual memory records to the teacher prompt.

Selection is deterministic rather than nearest-neighbor retrieval. The provider first selects an environment-level guide and then

Table 4: Benchmark and evaluation details for the three agent environments.

Benchmark	Interaction	Development / evaluation use	Primary metric	Max turns
ALFWorld	Text household control	128 validation tasks	Success (%)	50
WebShop	Simulated web shopping	128 tasks/checkpoint from goals 0–499	Score and success (%)	15
Search	Search-augmented QA	504-example development; 51K validation	Seven-dataset macro EM (%)	4

appends one task-matched file. ALFWorld maps six household task types to five skill files; Search maps metadata to direct-retrieval, multi-hop, entity-attribute, or comparison guidance; WebShop maps the requested product to one of seven categories. Skill text shares the prompt budget (2,048 tokens for ALFWorld and 4,096 for WebShop/Search). The ordinary branch and inference-time policy never receive this text.

The teacher prefix is inserted verbatim as [Privileged Skill Information], followed by a newline, the selected skill, and two newlines before the ordinary environment prompt. ALFWorld and WebShop require reasoning inside `<think>` tags and one admissible action inside `<action>`; Search alternates `<search>` queries with `<information>` observations and emits either one search query or a final `<answer>` per turn.

A.5 Optimization and Named Configurations

Tables 5 and 6 collect the archived training configuration.

Table 5: Common optimization settings.

Setting	Value
Learning rate / weight decay	10^{-6} / 0.01
LR warmup ratio	0 (Search main 3B: 0.1)
PPO epochs / loss reduction	1 / token mean
Rollouts per prompt	8
Train / validation temperature	1.0 / 0.4 (sampling)
Numeric type	bfloat16
Gradient checkpointing	enabled
Actor / reference parameter offload	disabled / enabled
Teacher baseline / η mode	step / fixed
Teacher temperature T / gate τ	1.0 / 2.0
Score-normalization ϵ	10^{-8}
GateNorm default	disabled
Score normalization	global or per-sequence
GiGPO γ / step weight	0.95 / 1.0
GiGPO mode	mean normalization
Anchor similarity / threshold	disabled / 0.95

All ADRS models are trained on multiple NVIDIA H100 GPUs. We follow the SDAR experimental configuration [15] for the benchmark interface, rollout schedule, and evaluation protocol, while replacing its optimization objective with the proposed reward-shaping framework.

ADRS-global uses step centering and batch-global scaling; *ADRS-per-seq* changes only score scaling to per-sequence normalization in the matched fix3/fix4 comparisons. *GateNorm* applies Eq. 4.2; *ADRS (no TVA)* uses the corresponding environment configuration with $m = 1$. The primary ALFWorld row uses GRPO, L2+GateNorm, $\eta = 0.1$, entropy coefficient 0.01, asymmetric

Table 6: Primary optimization and environment settings.

Setting	ALFWorld	WebShop	Search
Train batch	16	16	128
Eval batch	128	128	1024
Max turns	50	15	4
Prompt length	2048	4096	4096
Response length	512	512	512
Truncation	error	error	left
PPO mini-batch	256	64	256
Micro-batch / GPU	16	8	16
Invalid-action penalty	0.1	0.1	0.01
η	0.1	0.02	0.05–0.1
KL coefficient	0.01	0.01	0.001

clip bounds 0.2/0.28, and gradient clipping at 1.0. The primary WebShop row uses GRPO, L2 per-sequence normalization, and $\eta = 0.02$. The strongest 3B Search row at step 150 is the global-normalized $\eta = 0.05$ run. The ALFWorld no-TVA/TVA rows in the component sequence instead share GiGPO and common optimization settings except the gate.

A.6 TVA Levels and Non-Return Fallback

The implementation chooses one available level for an entire run; it does not fall back separately for each undersized group.

- **Step-level TVA (L3):** units are repeated GiGPO anchor states, and R_u is the pre-ADRS environment return-to-go.
- **Completion-level TVA (L2):** units are trajectories sampled for the same prompt, and R_u is the pre-ADRS completion return.
- **Token fallback (L1):** when reference log-probabilities exist but L2/L3 tensors do not, the implementation uses

$$m_t^{L1} = \text{sg}\left[\sigma\left(\tau\left(\ell_t^T - \ell_t^{\text{ref}}\right)\right)\right], \quad (15)$$

where ℓ_t^{ref} is the detached frozen-reference score without skill text. This fallback contains no return and is not a TVA estimator.

At L2/L3, a group with fewer than two units is assigned $d_g = 0$ and hence $m = 0.5$; it does not dynamically fall back to another level. Turning TVA off is different: it sets $m = 1$.

A.7 Canonical Training Procedure

Algorithm 1: ADRS training iteration. **Input:** behavior policy π_{θ_b} , task batch, skill provider ρ , teacher scale η , selected gate level, and backbone $\mathcal{B}$. **Output:** updated actor parameters θ .

- (1) Sample K ordinary-context trajectories per task from π_{θ_b} and record valid-token masks, environment outcomes, and the backend’s base token rewards.
- (2) With θ_b frozen, compute ordinary scores ℓ^b and skill-conditioned scores ℓ^T on the same sampled tokens. Treat

both score tensors and all subsequent teacher-branch quantities as detached.

- (3) Compute the step-centered and normalized teacher score $\hat{q}$ with Eq. 2.
- (4) If TVA is enabled, compute the selected L2/L3 return contrast and detached gate with Eqs. 4–4.2; if L1 is selected, use Eq. 15; otherwise set $m = 1$.
- (5) Form and save the teacher token reward $r^T = \eta m \hat{q}$, and form the shaped token reward $\tilde{r} = r^{\text{base}} + r^T$ as in Eq. 7.
- (6) **Trajectory path:** use the backbone’s native credit construction on $\tilde{r}$ to obtain A^{traj} . For the step-shared no-TVA/L2/L3 cases, the teacher reward has zero within-step sum, so this trajectory score equals the base-reward trajectory score. For GiGPO, retain its native step-level credit branch unchanged.
- (7) **Token path:** whiten the saved r^T over valid tokens within each prompt group g to obtain the detached token modulation $Z_t^T = (r_t^T - \mu_g^T) / \sqrt{(\sigma_g^T)^2 + \epsilon}$.
- (8) Merge the two paths once at token level, $A_{i,k,t}^{\text{ADRS}} = A_{i,k}^{\text{traj}} + \eta Z_{i,k,t}^T$, and detach the combined coefficient. For GiGPO, add ηZ^T to the episode-level advantage before combining it with the unchanged native step-level credit.
- (9) Optimize Eq. 12 with the backend’s configured clipping, KL, entropy, and gradient-control terms unchanged. Within the ADRS actor term, only the current actor log-probability receives gradient; native entropy, KL, and other regularizers retain their configured gradients.
- (10) Commit the updated actor as the behavior policy for the next iteration.

B Proofs and Scope of the Identities

Proof of Proposition 2. By the definition of b_s , $\sum_{j \in \mathcal{V}_s} q_{s,j} = \sum_j \ell_{s,j}^T - |\mathcal{V}_s| b_s = 0$. Division by a positive scalar shared within the step preserves the zero sum, as does multiplication by the step-shared ηm_s ; hence $\sum_{j \in \mathcal{V}_s} r_{s,j}^T = 0$. For any ordering $(j_1, \dots, j_{n_s})$, define $\Phi_{s,1} = 0$ and $\Phi_{s,r+1} = \Phi_{s,r} + r_{s,j_r}^T$. Then $r_{s,j_r}^T = \Phi_{s,r+1} - \Phi_{s,r}$ directly, and the endpoint is $\Phi_{s,n_s+1} = \sum_{r=1}^{n_s} r_{s,j_r}^T = 0$. The claim covers the canonical no-TVA, L2, and L3 cases when the gate is constant over the same valid-token set used for centering. It preserves only the unweighted shaping mass on that sampled step; it is neither a discounted-return guarantee nor a fixed, path-independent Markov potential.

Proof of Proposition 1. Write $\bar{R}_g = \mathbb{E}_g[R]$. With $\epsilon = 0$,

$$\begin{aligned} d_g &= \frac{\mathbb{E}_g[\alpha R]}{\bar{\alpha}_g} - \frac{\mathbb{E}_g[(1 - \alpha)R]}{1 - \bar{\alpha}_g} \\ &= \frac{\mathbb{E}_g[\alpha R] - \bar{\alpha}_g \bar{R}_g}{\bar{\alpha}_g(1 - \bar{\alpha}_g)} = \frac{\text{Cov}_g(\alpha, R)}{\bar{\alpha}_g(1 - \bar{\alpha}_g)}. \end{aligned}$$

The denominator is positive under the stated condition. GateNorm either divides by another positive scalar or leaves d_g unchanged, and the sigmoid with $\tau > 0$ is strictly increasing, proving the sign statement. With finite ϵ , the implemented contrast is a stabilized approximation to the covariance identity. Its sign and magnitude

approach Eq. 6 when both group-weight sums dominate ϵ ; no exact sign equivalence is claimed in the degenerate regime.

Proof of Proposition 3. Under the exact old-logprob condition $\ell_t^b = \log \pi_{\theta_b}(y_t | c_t)$ from Proposition 3, at θ_b we have $\omega_t = 1$ and $\nabla_{\theta} \omega_t = \omega_t \nabla_{\theta} \log \pi_{\theta} = \nabla_{\theta} \log \pi_{\theta}$. The valid-token reduction, samples, A_t^0 , ΔA_t , and a_t are fixed, so direct differentiation gives Eqs. 13 and 14. Setting $a_t = \Delta A_t$ makes their score-weighted gradients equal at that evaluation point.

The gradient equality is local. Away from the behavior-policy point, the ADRS coefficient is multiplied by ω_t whereas the auxiliary log-likelihood coefficient is not, and PPO clipping can separate them further. The result excludes full-vocabulary KL/JSD, representation matching, non-detached gates, post-combination advantage re-normalization, and an entire multi-epoch PPO update. It does not commute f through $\text{Adv}_{\mathcal{B}}$: ADRS’s coefficient is explicitly $\Delta A = \text{Adv}_{\mathcal{B}}(r^{\text{base}} + f) - \text{Adv}_{\mathcal{B}}(r^{\text{base}})$.

C Supplementary Results

C.1 Teacher Scale and Score Normalization

Table 7 reports the Search development-set sweep over the teacher scale η . The best standard-checkpoint and late-training settings differ: $\eta = 0.05$ is strongest at step 150, whereas $\eta = 0.1$ reaches the highest observed late checkpoint. The $\eta = 0.005$ run peaks early and falls to 9.6 at step 150, showing that an excessively weak teacher coefficient can produce unstable checkpoint behavior rather than a smooth reduction toward the outcome-only baseline.

Table 7: Search development-set sensitivity to η for ADRS-global+GateNorm.

η	Step 150	Peak	Peak step	Behavior
0.10	36.8	42.1	285	Strong late rise
0.05	36.9	38.1	135	Best at standard point
0.02	35.2	40.3	270	Slower late rise
0.005	9.6	38.3	90	Early peak, collapse

Table 8 compares global and per-sequence score normalization in matched fix3/fix4 pairs. Global normalization is stronger on ALFWorld and full Search, while per-sequence normalization is stronger on WebShop. This environment dependence explains why the main table uses benchmark-specific ADRS configurations rather than one universally selected normalization rule.

Table 8: Best observed percentages for matched global (fix3) versus per-sequence (fix4) normalization runs. Each pair changes only `normalize_mode`.

Environment / setting	Global	Per-seq	Higher
ALFWorld	90.6	82.8	Global
WebShop	79.7	82.0	Per-seq
Search, full	40.5	38.4	Global

C.2 Complete Long-Horizon and Ablation Matrices

Table 9 expands the representative 300-step results in the main text to every named variant. Best, final, and last-three values distinguish peak attainment from late-training persistence. Table 10 similarly separates fixed-step comparisons from peak and trajectory summaries, clarifying the contributions of ADRS, TVA, and the underlying backbone.

Optimization dynamics. The selected ALFWorld ADRS curve crosses 70% at approximately step 70, while the published SDAR curve crosses that threshold near step 130, corresponding to a 1.9× faster rise in terms of optimizer steps. Entropy trajectories also distinguish stable and aggressive variants: ADRS-global remains near 0.6 and ADRS+TVA ends near 0.34, whereas the entropy/clip variant rises to 8.63 and becomes unstable.

C.3 Action–Object Token Diagnostic

Table 11 provides the exact values underlying Figure 6. The action-to-other magnitude ratio rises from 1.6 at initialization to 95.3 at step 150 and remains high at step 250. After this stage, task performance approaches saturation and the policy changes more slowly, which explains the shift in the object term and the smaller ratio at the final checkpoint.

C.4 Training Dynamics across Models and Domains

Figures 9–10 show the full 3×3 grid across Qwen2.5-3B, Qwen2.5-7B, and Qwen3-1.7B on ALFWorld, Search, and WebShop. Success generally rises over training, but the interaction statistics differ by environment: ALFWorld and WebShop often shorten episodes as success improves, whereas Search remains constrained to a small number of tool turns. Together, these trajectories show consistent optimization progress across model families and agentic domains.

Table 9: Complete 300-step result matrix. “Best” is the highest observed checkpoint of the named run; Search uses the full 51K evaluation at step 300. Rows characterize named variants rather than one universal configuration.

Environment	Configuration	Best	Final	Last-3	Interpretation
ALFWorld	GRPO	92.2	86.7	86.2	Local outcome-only baseline
	GiGPO	96.9	91.4	93.0	Strong step-level baseline
	ADRS (no TVA), GiGPO	96.1	92.2	91.9	Fixed teacher scale
	ADRS+TVA, GiGPO	96.9	94.5	94.3	Stronger reported late checkpoints
	ADRS-global+GateNorm, GRPO	96.1	89.8	90.6	Primary step-150 system
	ADRS-global+entropy/clip, GRPO	97.7	96.1	96.1	Strongest late-stage run
WebShop	GRPO	78.9	78.9	75.3	Local outcome-only baseline
	ADRS (no TVA)	82.8	82.8	80.2	Fixed teacher scale
	ADRS-per-seq (primary)	82.0	78.9	77.3	Step-150 primary run
	ADRS-per-seq (late)	84.4	84.4	77.9	Highest final checkpoint
	ADRS-per-seq+GateNorm	84.4	82.0	78.9	Peak at step 275

Table 10: Complete ablation and backbone matrix. Last-3 is the mean of the final three reported checkpoints.

Environment	Evaluation	Factor	Reference	Ref.	Variant	Delta
<i>Fixed-step local comparisons</i>						
ALFWorld	Step 150	ADRS on GiGPO	GiGPO	82.8	ADRS+TVA 89.8	+7.0
ALFWorld	Step 150	Centered ADRS	GiGPO	82.8	ADRS (no TVA) 87.5	+4.7
ALFWorld	Step 150	TVA	ADRS (no TVA)	87.5	ADRS+TVA 89.8	+2.3
ALFWorld	Step 150	Primary system	Local GRPO	84.4	Primary ADRS 94.5	+10.1
WebShop	Step 150	ADRS system	GRPO	63.3	ADRS-per-seq 76.6	+13.3
Search	Step 150	ADRS	GRPO	38.4	ADRS-global 38.3	-0.1
<i>Descriptive peak and trajectory summaries</i>						
ALFWorld	Peak	Backbone	GRPO	92.2	GiGPO 96.9	+4.7
ALFWorld	Peak	ADRS on GRPO	GRPO	92.2	ADRS-entropy/clip 97.7	+5.5
ALFWorld	Peak	TVA	ADRS (no TVA)	96.1	ADRS+TVA 96.9	+0.8
ALFWorld	Last-3	TVA	ADRS (no TVA)	91.9	ADRS+TVA 94.3	+2.4
WebShop	Peak	ADRS	GRPO	78.9	ADRS (no TVA) 82.8	+3.9
WebShop	Peak	GateNorm	ADRS-per-seq	82.0	+GateNorm 84.4	+2.4
Search	Step 300	ADRS-global	GRPO	40.2	ADRS-global 40.5	+0.3

Table 11: Theory-oriented token diagnostic on four fixed ALFWorld trajectories. Values are signed log-probability differences, not KL terms.

Step	Action	Object	Other	Action / Other
0	-0.2509	0.3089	0.1552	1.6
50	-0.5155	0.4150	0.1037	5.0
100	-0.5563	0.2354	0.0539	10.3
150	-0.8582	0.4591	0.0085	95.3
200	-0.9665	0.3516	0.0373	26.1
250	-0.9089	0.4475	0.0229	39.5
300	-1.3105	-0.3664	0.5108	2.6

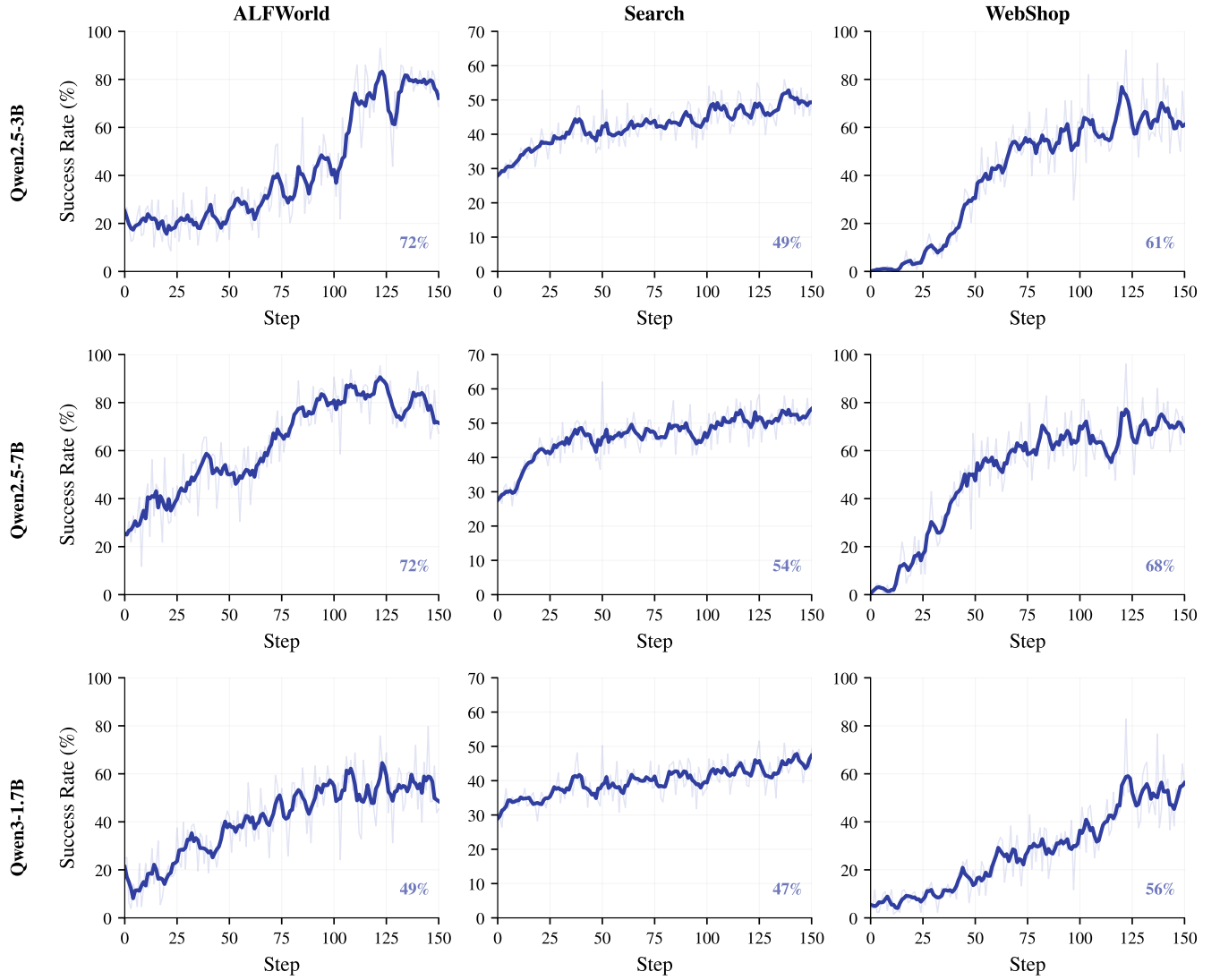


Figure 9: ADRS success-rate dynamics across three environments and three model configurations. Light lines are checkpoint traces and dark lines are smoothed trends.



Figure 10: Episode- and response-length dynamics across three environments and three model configurations. Top: mean episode length. Bottom: mean response length. Light lines show checkpoint traces, and dark lines show smoothed trends.