RECONCILING PROCESS SUPERVISION WITH OUTCOME-BASED CREDIT IN AGENTIC POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

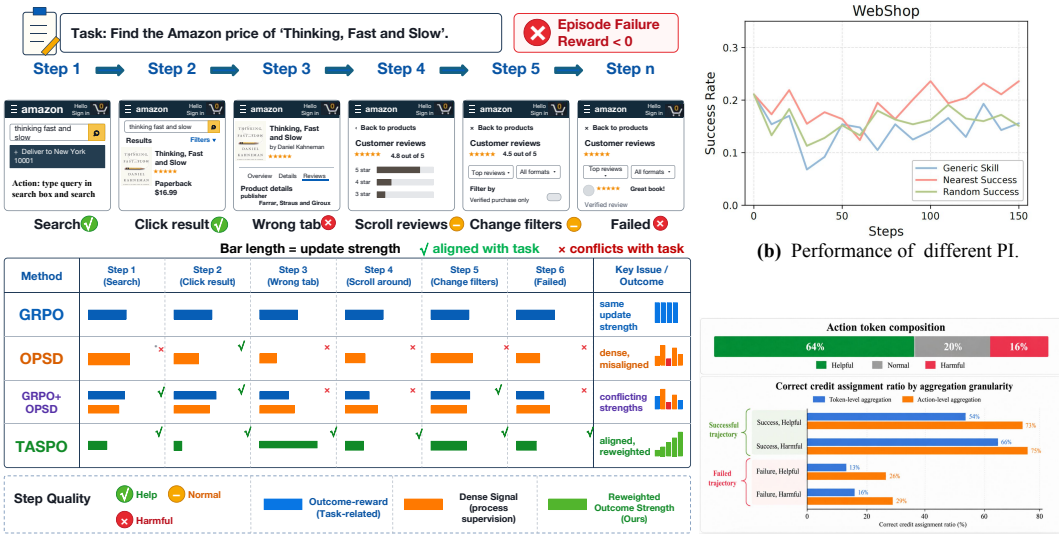
For language-model agents, outcome-based reinforcement learning provides verified task feedback but assigns the same trajectory-level advantage to every intermediate decision, making it difficult to identify which actions contributed to success or failure. Process supervision can refine this credit, yet treating it as an independent auxiliary objective may encourage or discourage intermediate decisions independently of the trajectory's outcome advantage. Self-distillation with privileged information provides finer-grained guidance through changes in response likelihood. However, these changes do not directly specify how to allocate outcome credit, and the guidance may depend on conditions absent from the target trajectory. We introduce **TASPO**, a framework for **constrained credit redistribution**. Outcome feedback sets the trajectory-level advantage; process supervision redistributes it across decisions while preserving its sign and mean within each trajectory. TASPO aligns guidance from verified successful sibling trajectories with the completed target interaction, measures guidance-induced support over executable actions, and converts these scores into positive, bounded, mean-one weights. The constraints preserve advantage polarity and decision-level mean while limiting local reweighting, without additional environment interaction. Across ALFWorld, Search-QA, and WebShop with three backbone models, TASPO improves over GRPO by 12.4 percentage points on average. Analyses confirm that both alignment and action-level aggregation are essential, while TASPO stabilizes policy-KL optimization dynamics. Codes are available at https://anonymous.4open.science/r/TASPO-0DAA/.

1 INTRODUCTION

Reinforcement learning enables language-model agents to solve complex, multi-turn tasks through interaction with external environments (Jin et al., 2025; Feng et al., 2025a; Wang et al., 2025). In sparse-reward settings, outcome-based methods such as GRPO propagate a trajectory-level group-relative advantage uniformly to all decisions within a trajectory (Shao et al., 2024). This creates a fundamental credit-assignment bottleneck: task-critical choices and avoidable mistakes receive identical reinforcement signals. For example, an agent may unsuccessfully try to open a cabinet, move to the cabinet, and then successfully retry. Uniform trajectory-level credit assigns the same advantage to the ineffective attempt and the recovery actions. Process supervision offers a natural way to differentiate intermediate decisions, but using it for outcome-based reinforcement requires more than a denser training signal.

Existing approaches obtain finer credit through environment-state structure, learned process evaluators, or value estimates (Lightman et al., 2024; Feng et al., 2025b; Choudhury, 2025). On-policy (self)-distillation (OPSD) offers another source of process evidence: a teacher branch conditions on training-time privileged information (PI), such as successful execution context, to re-evaluate responses sampled without PI (Agarwal et al., 2024; Zhang et al., 2026; Zhao et al., 2026; Lu et al., 2026; Yang et al., 2026c). The resulting likelihood shifts reveal how additional context changes local policy preferences without a separately trained process reward model. Turning these preferences into outcome-consistent credit, however, exposes two fundamental mismatches.

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.



(a) An explanation of the optimization conflict between outcome supervision and dense process supervision.

(b) Performance of different PI.

(c) Action-level vs. Token-level Credit Assignment

Figure 1: **Motivation for constrained credit redistribution. (a)** Directly combining process and outcome objectives can produce misaligned updates. **(b)** Self-distillation performance depends on the source and relevance of PI. **(c)** Action-level aggregation achieves higher credit-assignment accuracy than token-level shifts.

First, process preference shifts encode relative likelihood changes rather than outcome credit. PI-induced likelihood shifts describe how a policy changes under additional information; an outcome-derived advantage instead determines whether behavior associated with the realized trajectory should be reinforced or suppressed. Because the two signals encode different quantities, optimizing a distillation loss alongside the RL objective introduces a second target whose gradient is not constrained to align with outcome-based reinforcement (Figure 1(a)). Weighting or gating the auxiliary loss can regulate its influence (Lu et al., 2026; Zhang et al., 2026; Ding et al., 2026), but does not determine how verified trajectory-level credit should be distributed across decisions.

Second, process evidence faces mismatches in context and granularity. Guidance from successful rollouts depends on the states in which it was effective. When a target trajectory follows a different path, the same guidance can become irrelevant or misleading (Xu et al., 2026; Fu et al., 2026). During training, the completed target interaction provides the context for checking whether this guidance applies. Meanwhile, environments execute complete actions, but privileged guidance can produce different likelihood shifts across their tokens. Direct token-level weighting thus assigns different credit to tokens within the same executable decision. (Figure 1(b,c)). These mismatches motivate aligning guidance with the target context and aggregating process evidence over executable actions.

These challenges raise a central question: *how can process supervision refine local credit while keeping verified outcomes as the basis for reinforcement?* Our key insight is to separate their roles: *outcome feedback sets the sign and average credit within each trajectory, while process supervision determines its relative allocation across decisions.* We use process evidence to reweight the outcome-derived advantage rather than add a separate distillation objective.

Based on this principle, we introduce **TASPO**, a framework for **constrained credit redistribution**. For a trajectory-level advantage A, TASPO assigns decision t a redistributed advantage $A_t = w_t A$, with positive, bounded, mean-one weights. Positivity preserves the polarity of the outcome signal; mean-one normalization preserves its average decision-level credit within the trajectory; and boundedness limits excessive local amplification or suppression. TASPO obtains these weights by aligning guidance from verified successful sibling trajectories with the completed target interaction. A frozen rollout policy then scores each sampled response with and without the aligned guidance and aggregates the likelihood shift over the executable action to estimate relative support. The resulting weights are applied uniformly to the generated tokens associated with each decision, requiring no additional environment interaction.

We evaluate TASPO on ALFWorld (Shridhar et al., 2021), Search-QA (Jin et al., 2025), and WebShop (Yao et al., 2022) across three backbones. Across nine benchmark–backbone settings, TASPO outperforms GRPO by an average of 12.4 percentage points under matched interaction and update budgets. Controlled analyses show that trajectory-aligned guidance outperforms generic skills or raw trajectory reuse, while action-level redistribution improves task performance. On ALFWorld, TASPO also exhibits smoother reward and policy-KL dynamics, consistent with more stable optimization.

Our contributions are threefold:

1. We reframe the synergy between process supervision and outcome feedback as *constrained credit redistribution*, decoupling outcome semantics from decision-level allocation.
2. We propose **TASPO**, an on-policy method that aligns process guidance to the realized target interaction, aggregates support over executable actions, and rescales outcome advantages via positive, bounded, mean-one weights.
3. Across nine benchmark–backbone configurations, TASPO consistently outperforms GRPO by 12.4 percentage points on average; controlled analyses isolate the distinct benefits of supervision alignment, action-level allocation, and constrained optimization.

2 Related Work

2.1 Agentic Reinforcement Learning and Credit Assignment

Outcome-based training provides verified trajectory feedback but coarse intermediate credit (Shao et al., 2024; Jin et al., 2025; Wang et al., 2025). Finer credit can require repeated states in GiGPO (Feng et al., 2025b), reference-answer likelihoods in A^{2}TGPO (Chen et al., 2026), or branched continuations in AT^{2}PO and APPO (Zong et al., 2026; Wang et al., 2026b). These requirements constrain available supervision or change rollout collection. ProGPO's progress-based exploration (Yang et al., 2026b) and SAPO's KL regularization (Li et al., 2026b) address uninformative rewards and unstable updates, but do not directly resolve how process guidance should redistribute existing outcome credit. TASPO aligns guidance from successful siblings with recorded target paths and converts relative action-level support into positive, bounded, mean-one weights, preserving outcome-derived advantage signs and their trajectory mean without additional environment rollouts.

2.2 On-Policy Self-Distillation with Privileged Information

On-policy distillation supervises student-generated samples with a teacher model, reducing distribution mismatch (Agarwal et al., 2024). Privileged contexts provide denser supervision (Zhao et al., 2026; Ye et al., 2026; Yang et al., 2026a; Lu et al., 2026; Wang et al., 2026a; Yang et al., 2026c; Wu et al., 2026), but their guidance can depend on source-specific states or action preconditions absent from the target's realized path. Step-aware, sibling-guided, and counterfactual methods improve supervision granularity and grounding (Zhang et al., 2026; Ding et al., 2026; Meng & Chen, 2026). Finer supervision alone does not establish how to allocate outcome credit across intermediate decisions. Directly optimizing process preferences can also introduce gradients that compete with outcome-based reinforcement. Reliability analyses further examine teacher compatibility, token importance, trajectory position, curriculum, and trust-region constraints (Fu et al., 2026; Li et al., 2026c; Xu et al., 2026; Xie et al., 2026; Li et al., 2026a; Xing et al., 2026). TASPO addresses applicability by checking source-supported guidance against completed target histories. It aggregates likelihood shifts over executable actions to avoid fragmented token-level credit, then uses relative support to redistribute the verified outcome advantage. Positive, bounded, mean-one weights retain outcome-derived signs and average credit while incorporating process guidance without a separate distillation objective.

3 Method

TASPO uses process supervision not as an independent objective, but to redistribute verified outcome credit. Terminal feedback anchors advantage polarity and trajectory-average credit, while privileged process evidence determines its relative allocation across decisions. Figure 2 summarizes the framework, including evidence alignment and action-level credit redistribution.

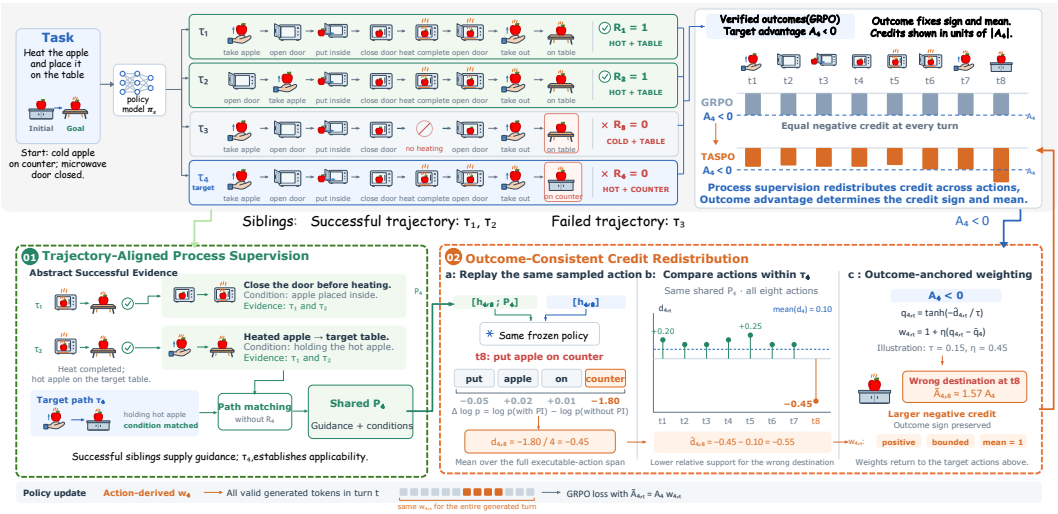


Figure 2: **Overview of TASPO.** Successful sibling trajectories provide candidate guidance, which is checked against the completed target history (lower left). The frozen rollout policy then evaluates the sampled responses with and without the shared guidance, aggregates likelihood shifts over executable actions, and converts relative action scores into positive, bounded, mean-one weights (lower right). Each weight rescales the outcome advantage for all generated tokens in the corresponding turn.

3.1 Outcome-Constrained Credit Assignment

For task x, the rollout policy $\pi_{\theta_{\text{old}}}$ samples K trajectories $\mathcal{G}_x = \{\tau_i\}_{i=1}^{K}$. At turn t, it generates response y_{it} from history h_{it}, from which the environment executes action a_{it}. Terminal reward R_i yields the GRPO advantage $A_i = (R_i - \mu_{\mathcal{G}_x})/(\sigma_{\mathcal{G}_x} + \delta)$, where $\mu_{\mathcal{G}_x}$ and $\sigma_{\mathcal{G}_x}$ are group reward statistics and $\delta > 0$ stabilizes normalization. Standard GRPO assigns the same A_i to all T_i decisions, leaving their distinct contributions unresolved.

TASPO instead redistributes A_i across turns:

$$\widetilde{A}_{it} = A_i w_{it}, \qquad 1 - \epsilon_w \le w_{it} \le 1 + \epsilon_w, \qquad \frac{1}{T_i}\sum_{t=1}^{T_i} w_{it} = 1, \tag{1}$$

with $0 \le \epsilon_w < 1$. Positive weights preserve the sign of every decision's advantage, mean-one normalization preserves the average advantage across turns, and boundedness limits each adjustment to $\epsilon_w|A_i|$. Thus, process supervision changes the strength of reinforcement across decisions while retaining the original outcome advantage's sign and mean.

3.2 Trajectory-Aligned Process Supervision

We construct guidance applicable to the target trajectory's realized interaction path. **Candidate extraction.** A training-time analyzer extracts guidance supported by actions and observations from verified successful trajectories $\mathcal{E}_x$ from verified successful trajectories $\mathcal{S}_x \subseteq \mathcal{G}_x$, including task constraints, action preconditions, and valid alternative strategies. Each item cites its supporting source actions and observations. For target τ_i, items supported solely by that target are excluded, yielding $\mathcal{E}_{x,-i}$. This construction ensures that every retained item has supporting evidence from at least one successful trajectory other than the target. **Target-history alignment.** Because successful-trajectory guidance can depend on state-contingent preconditions, the analyzer checks each candidate against task x and the completed target history $\mathcal{H}_i$, retaining only situationally consistent guidance:

$$P_i = \mathcal{A}_{\text{align}}(x, \mathcal{H}_i, \mathcal{E}_{x,-i}), \tag{2}$$

where $\mathcal{A}_{\text{align}}$ denotes the training-time analyzer call and P_i the resulting privileged context. This prevents sibling-specific guidance from being transferred to incompatible states along the target trajectory. Source interactions ground the guidance, while target history establishes its relevance; no explicit terminal reward label is supplied during alignment. We reuse P_i across turns so that all

action scores are computed with a fixed guidance set. Differences between turns therefore reflect how that guidance changes the likelihood of their sampled actions, rather than changes in the guidance itself. If no candidate passes alignment, $P_i = \emptyset$ and TASPO falls back to GRPO. Appendix A.2 gives implementation details.

3.3 Relative Action Scoring

Conditioned on P_i, we measure how guidance changes policy support for each sampled decision. Because environments execute complete actions rather than individual tokens, TASPO scores only the executable action span $\mathcal{M}_{it}$. Using the frozen rollout policy, we compute

$$\Delta_{itk} = \mathrm{sg}[\log \pi_{\theta_{\mathrm{old}}}(y_{itk} \mid P_i, h_{it}, y_{it,<k}) - \log \pi_{\theta_{\mathrm{old}}}(y_{itk} \mid h_{it}, y_{it,<k})], \tag{3}$$

where sg stops gradient propagation. Positive Δ_{itk} indicates increased support under privileged guidance. Let $\mathcal{I}_i$ denote scorable turns with complete action spans and finite log-probabilities.

We clip token shifts by $c_\Delta > 0$ to reduce outlier sensitivity, average them over each action, and center scores across scorable turns:

$$\begin{aligned} d_{it} &= \frac{1}{|\mathcal{M}_{it}|} \sum_{k \in \mathcal{M}_{it}} \mathrm{clip}(\Delta_{itk}, -c_\Delta, c_\Delta), \\ \widehat{d}_{it} &= d_{it} - \frac{1}{|\mathcal{I}_i|} \sum_{u \in \mathcal{I}_i} d_{iu}, \qquad t \in \mathcal{I}_i. \end{aligned} \tag{4}$$

Thus $\widehat{d}_{it} > 0$ means that the action has a larger clipped log-likelihood shift than the average across scorable turns. TASPO uses this within-trajectory comparison to determine relative credit weights.

3.4 Advantage Redistribution and Policy Optimization

Credit redistribution is sign-asymmetric: a better-supported decision should receive more positive credit when $A_i > 0$, but less negative credit when $A_i < 0$. TASPO therefore orients support by the outcome sign:

$$q_{it} = \tanh\left(\frac{\mathrm{sign}(A_i)\widehat{d}_{it}}{\tau}\right), \qquad t \in \mathcal{I}_i, \tag{5}$$

for $A_i \neq 0$ and $|\mathcal{I}_i| \geq 2$, where $\tau > 0$ controls sensitivity. Multiplying by $\mathrm{sign}(A_i)$ reverses the mapping from support to weight when the advantage is negative. Consequently, greater support leads to more favorable credit under either sign: stronger reinforcement when $A_i > 0$ and a smaller penalty when $A_i < 0$.

We convert these scores into zero-sum deviations:

$$w_{it} = 1 + \eta(q_{it} - \bar{q}_i), \qquad \bar{q}_i = \frac{1}{|\mathcal{I}_i|} \sum_{u \in \mathcal{I}_i} q_{iu}, \qquad \eta = \frac{\epsilon_w}{2}. \tag{6}$$

Since $q_{it} \in [-1, 1]$, this construction guarantees $w_{it} \in [1 - \epsilon_w, 1 + \epsilon_w]$ and mean-one weights. Hence $\mathrm{sign}(\widetilde{A}_{it}) = \mathrm{sign}(A_i)$, $\frac{1}{T_i}\sum_t \widetilde{A}_{it} = A_i$, and $|\widetilde{A}_{it} - A_i| \leq \epsilon_w |A_i|$. Unscorable turns, or cases with $A_i = 0$ or $|\mathcal{I}_i| < 2$, use $w_{it} = 1$. Appendix B.1 provides the derivation.

Scoring and optimization use different token spans by design: $\mathcal{M}_{it}$ isolates the executable action for support estimation, whereas $\mathcal{R}_{it}$ denotes the full generated response receiving the resulting turn-level credit. This separates the decision evidence used for scoring from the tokens receiving its resulting credit. The policy objective is:

$$\mathcal{L}_{\mathrm{policy}} = \frac{1}{B}\sum_{i=1}^{B} \frac{1}{T_i}\sum_{t=1}^{T_i} \frac{1}{|\mathcal{R}_{it}|} \sum_{k \in \mathcal{R}_{it}} \ell_{itk}^{\mathrm{GRPO}}(\theta; \widetilde{A}_{it}), \tag{7}$$

where B is the trajectory batch size and ℓ^{GRPO} is the clipped surrogate loss. Turn-wise token averaging gives each decision equal base weight before redistribution, so mean-one normalization corresponds directly to preserving trajectory-average scalar credit. TASPO and all rerun baselines use the same normalization. Appendices B.4 and B.5 provide gradient-perturbation and finite-sample analyses.

Table 1: Main results on ALFWorld, Search-QA, and WebShop across three backbone models. ALFWorld reports task-family success rates, Search-QA reports dataset accuracy and WebShop reports Score and success rate (Succ.), all in percent. Avg. is the unweighted mean of the six ALFWorld task-family scores or the seven Search-QA dataset scores. **Bold** and underlined values mark the best and second-best distinct values within each backbone.

	ALFWorld							Search-QA								WebShop	
Method	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	19.7	24.6	48.1	31.0	26.3	25.3	7.2	59.7	31.7	6.7	0.8
OPSD	48.8	41.7	16.7	0.0	15.8	16.7	23.3	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	11.3	3.1
GRPO	87.9	68.8	81.8	58.3	65.4	68.4	71.8	39.3	60.6	41.1	37.4	34.6	15.4	26.4	36.4	79.8	63.3
GiGPO	94.3	72.7	93.8	90.1	64.3	89.5	84.1	43.0	60.5	**46.9**	36.2	35.5	12.1	64.1	42.6	83.8	70.3
SDAR	97.1	62.5	**100.0**	61.9	75.0	84.2	80.1	44.8	58.1	44.3	38.6	36.2	15.7	66.1	43.4	85.0	68.0
StepOPSD	97.1	66.7	87.0	79.1	78.9	**95.0**	84.0	43.6	61.2	43.8	39.2	38.1	15.8	64.5	43.7	–	–
OPID	92.7	**100.0**	88.9	70.0	**84.2**	70.0	84.3	45.9	61.4	45.7	40.7	**38.8**	16.4	66.1	45.0	85.0	74.2
TASPO	**97.9**	77.5	95.8	**95.8**	79.6	70.8	**86.2**	**46.8**	**65.5**	44.9	**41.1**	37.4	**19.1**	**70.2**	**46.4**	**88.5**	**78.1**
Qwen2.5-7B-Instruct																	
Vanilla	36.1	22.2	3.1	0.0	0.0	0.0	10.2	25.2	50.8	29.5	29.0	29.0	10.4	63.7	33.9	5.9	1.6
GRPO	91.2	70.0	90.9	78.6	64.7	66.7	77.0	46.3	63.8	47.2	44.6	44.3	19.3	71.4	48.1	85.7	75.0
GiGPO	97.4	82.3	95.8	83.7	**86.9**	78.7	87.5	46.3	64.1	46.5	41.6	43.6	18.9	68.9	47.1	84.4	72.8
OPSD	50.0	60.0	22.7	21.4	17.6	9.5	30.2	8.8	8.6	17.5	2.5	4.2	0.5	1.2	6.2	4.5	2.3
SDAR	94.7	75.0	**100.0**	86.7	68.2	78.9	83.9	46.3	63.5	48.2	43.8	**48.4**	19.6	73.0	49.0	**89.4**	**82.8**
OPID	**100.0**	81.8	97.1	**100.0**	80.8	80.0	90.0	48.8	65.6	46.8	**46.1**	42.7	**21.7**	72.6	49.2	85.3	79.7
TASPO	98.2	**84.8**	94.4	98.3	83.3	**80.7**	**90.1**	**49.6**	**65.9**	**49.3**	45.1	43.3	18.4	**76.7**	**49.8**	87.7	76.4
Qwen3-1.7B-Instruct																	
Vanilla	25.0	22.2	3.1	0.0	21.4	4.2	12.7	29.4	46.9	37.0	23.5	19.6	6.4	10.5	24.8	46.5	4.7
OPSD	26.3	33.3	9.1	0.0	4.5	5.3	13.1	4.2	8.3	4.6	6.6	15.3	0.7	1.2	5.8	47.4	9.3
GRPO	59.3	46.7	12.5	**68.4**	16.7	36.8	40.1	43.0	59.7	**47.4**	37.9	37.4	13.6	66.1	43.6	55.8	43.0
GiGPO	**79.9**	60.3	62.6	62.5	59.1	**65.1**	64.9	41.1	59.3	45.1	34.1	32.7	10.5	64.5	41.0	70.0	45.3
SDAR	73.5	25.0	76.9	33.3	40.0	36.8	47.6	39.7	58.9	45.3	35.9	35.5	12.6	65.3	41.9	76.8	58.6
StepOPSD	64.7	44.4	56.5	60.9	42.1	55.0	53.9	40.5	58.6	45.6	34.9	29.8	10.8	64.5	40.7	–	–
OPID	65.9	**72.7**	66.7	40.0	63.2	45.0	58.9	38.1	58.1	43.4	35.5	31.7	11.7	64.5	40.4	**79.6**	64.8
TASPO	74.2	70.8	**78.4**	60.3	**73.3**	60.7	**69.6**	**46.1**	**62.6**	45.4	**40.5**	**43.7**	**19.7**	**68.5**	**46.6**	77.6	**67.1**

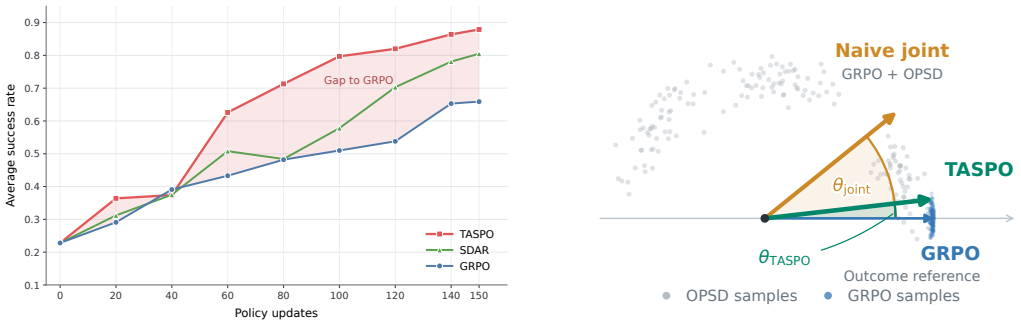
4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and evaluation. We evaluate TASPO across three diverse, long-horizon interactive agent domains: ALFWorld, WebShop, and Search-QA, strictly adhering to established agentic RL and distillation protocols (Lu et al., 2026; Zhang et al., 2026; Yang et al., 2026c; Wu et al., 2026). For ALFWorld, we track success rates across all six canonical task families, Pick, Look, Clean, Heat, Cool, and Pick2, reporting their macro-average. Primary evaluations utilize the 140-task seen split, alongside matched reruns on the 134-task unseen split. For WebShop, we benchmark over 128 fixed test environments, tracking normalized scores and task success rates. Search-QA spans seven multi-hop and open-domain datasets; we train policies on NQ and HotpotQA and report exact-match accuracy across all benchmarks.

Baselines. We compare against the instruction-tuned backbone without post-training(Vanilla), GRPO, and recent methods using training-time skills or privileged supervision, including GIGPO, SDAR, StepOPSD, OPID (Feng et al., 2025b; Lu et al., 2026; Zhang et al., 2026; Yang et al., 2026c). For controlled comparisons, we rerun methods with the same backbone, environment-interaction budget, number of policy updates, and evaluation tasks. Results taken directly from prior work are reported separately for reference, while our main comparisons are based on matched reruns.

Training configuration. We assess empirical scaling across three architectures: Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, and Qwen3-1.7B-Instruct, designating Qwen2.5-3B as the default backbone for analytical ablations. For both TASPO and rerun baselines, we maintain eight sample rollouts per task across 150 optimization iterations. Task batch sizes are configured to 16 for ALFWorld and WebShop, and 128 for Search-QA, with maximum horizons of 50, 15, and 4 turns, respectively. All runs optimize with a learning rate of 10^{-6} averaged over three random seeds. Privileged guidance extraction and alignment leverage a frozen DeepSeek-V4-Pro analyzer, setting default



(a) Training progress on ALFWorld.

(b) Gradient alignment with outcome supervision.

Figure 3: **Training dynamics and optimization alignment. (a)** TASPO improves faster and reaches a higher final success rate than GRPO and SDAR under the same policy-update budget. **(b)** TASPO gradients remain more closely aligned with the outcome-supervised GRPO reference than gradients from directly combining GRPO and OPSD objectives.

sensitivity $\tau = 0.5$ and deviation bound $\epsilon_w = 0.4$. Appendices C and D.5 detail hyperparameters and computational overhead.

4.2 Main Results

Consistent gains across backbones. As shown in Table 1, TASPO systematically improves upon GRPO across all benchmarks and architectures. On ALFWorld, macro-averaged success increases substantially by 14.4, 13.1, and 29.5 percentage points across Qwen2.5-3B, Qwen2.5-7B, and Qwen3-1.7B, respectively. TASPO achieves the highest Search-QA macro-average among the compared methods for each backbone. With Qwen2.5-3B, it also exceeds the strongest competing ALFWorld macro-average and WebShop success rate by 1.9 and 3.9 percentage points, respectively. Together with the gains over GRPO in all nine settings, these results demonstrate the effectiveness of credit redistribution across the evaluated agent domains.

Faster policy improvement. Figure 3(a) shows that TASPO improves faster after the initial exploration stage and reaches a higher final success rate than GRPO and SDAR under the same policy-update budget. We attribute this to two observations. First, as visualized in Figure 3(b), directly combining auxiliary distillation with GRPO produces conflicting gradient updates, whereas TASPO's redistribution preserves outcome gradient alignment. By rescaling rather than replacing verified advantage, TASPO provides fine-grained updates without compromising reinforcement polarity. Second, TASPO constructs PI from verified successful siblings in the current rollout group. Early in training, few successful trajectories are available, so applicable guidance is limited and TASPO falls back to GRPO when none can be constructed. As the policy improves, successful rollouts provide richer evidence for aligned guidance, enabling progressively finer credit allocation. This creates a reinforcing cycle in which better policies yield more useful process evidence, which in turn supports further policy improvement. Appendix E illustrates error recovery and task-constraint handling in ALFWorld and WebShop.

4.3 Analysis of Privileged Information

PI construction matters. Table 2 compares PI construction strategies under the same training protocol. Successful trajectory reuse provides larger gains than generic skills, suggesting the value of concrete interaction evidence. TASPO's abstracted, trajectory-aligned PI further achieves 86.3% success, exceeding Nearest Success by 5.2%. Its construction combines conditional guidance extraction with target-trajectory matching, addressing source-specific decisions that may be inapplicable to the target's interaction path. These results support transforming successful experience into decision-applicable supervision before using it to refine outcome-based credit. Appendix D.1 provides further isolated ablations over alignment filtering modules.

Analyzer robustness. We assess sensitivity to the training-time extraction model while keeping downstream optimization fixed (Table C.4). Across three distinct analyzer backbones, validated PI coverage remains remarkably consistent (71.6%–72.4%), with final task success varying marginally between 85.7% and 86.2%. Across the three evaluated analyzers, final ALFWorld success varies by

Table 2: **PI construction on ALFWorld.** Macro-averaged success rates with Qwen2.5-3B-Instruct. Alignment checks whether source guidance applies to the target trajectory's recorded interaction context.

PI construction	Source	Alignment	ALF Avg. (%)
GRPO	–	–	71.8
Generic Skill	Task skill	×	76.9
Random Success	Successful trajectory	×	79.2
Nearest Success	Similar successful trajectory	×	81.1
TASPO	Guidance from successful siblings	✓	**86.2**

Table 3: **Effect of credit allocation granularity.** Results with Qwen2.5-3B-Instruct. ALF Avg. and Search Avg. are macro averages; performance scores are in percent. ALF Steps is the mean number of environment steps among successful ALFWorld trajectories.

Allocation unit	ALF Avg. ↑	Search Avg. ↑	WebShop Succ. ↑	ALF Steps ↓
Token-level	79.7 ± 2.9	42.9 ± 1.7	72.1 ± 3.9	11.4
Action-level	$\mathbf{86.3 \pm 1.4}$	$\mathbf{46.4 \pm 1.2}$	$\mathbf{78.1 \pm 2.6}$	**9.2**

only 0.5 percentage points under the same downstream optimization configuration, indicating limited sensitivity to analyzer choice in this setting.

4.4 Token-Level versus Action-Level Allocation

Action-level allocation improves performance. Table 3 compares credit allocation granularities under identical aligned-PI pipelines and computational budgets. Averaged over three random seeds, action-level allocation consistently outperforms token-wise assignment across all domains, yielding gains of 6.6 pp on ALFWorld, 6.0 pp on WebShop, and 3.5 pp on Search-QA. Notably, action-level allocation also reduces the across-seed standard deviation from 2.9 to 1.4 percentage points on ALFWorld. Among successful episodes, the mean number of environment steps decreases from 11.4 to 9.2. Appendix D.3 analyzes the complementary effect of propagating action weights across full reasoning responses.

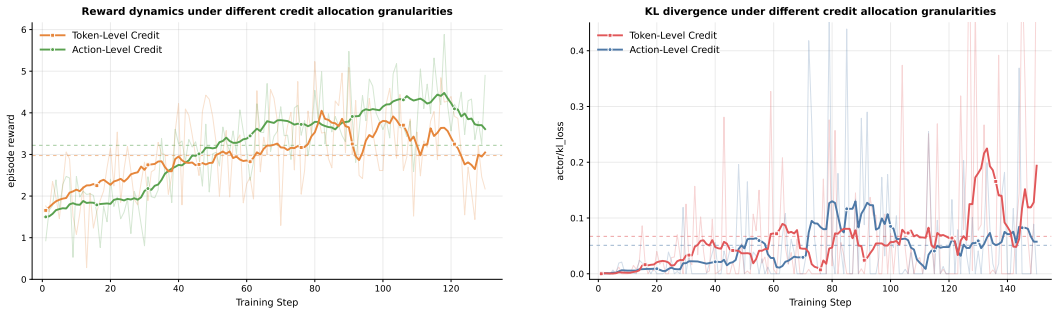
Steadier optimization dynamics. Figure 4 shows higher late-stage reward with action-level allocation and a larger late-stage KL increase in the token-level variant. Action-level scoring clips individual token shifts and averages them before assigning a shared turn weight, limiting the influence of any single token on credit allocation. These observations support aggregating process scores at the level of executable actions.

4.5 Credit Reweighting versus Distillation

Reweighting outperforms distillation. Table 4 evaluates alternative mechanisms for incorporating auxiliary supervision. Standalone OPSD severely degrades performance compared to GRPO (28.1% vs. 71.8%), while joint optimization (GRPO + OPSD) recovers performance to 81.7%. TASPO reaches 86.2%, outperforming both joint loss training and the strongest external teacher baseline (OPD with Qwen3-32B) by 4.5% and 2.6%, respectively. These empirical margins directly corroborate our gradient analysis in Figure 3(b): unconstrained auxiliary objectives introduce competing gradient directions that dilute task reinforcement. By reparameterizing privileged evidence into bounded advantage weights, TASPO enriches intermediate feedback

Table 4: **Supervision strategies on ALFWorld.** Macro-averaged success rates with Qwen2.5-3B-Instruct. Self + PI denotes the PI-conditioned rollout policy used for scoring.

Method	Teacher	ALF Avg. (%)
GRPO	–	71.8
OPSD	Self	28.1
OPD	Qwen2.5-7B	76.4
OPD	Qwen3-32B	83.6
GRPO + OPSD	Self	81.7
TASPO	Self + PI	**86.3**



(a) **(b)**

Figure 4: **Optimization dynamics under different credit allocation granularities.** (a) Action-level allocation achieves smoother reward improvement and a higher final reward compared with token-level allocation. (b) Token-level allocation introduces larger policy-update fluctuations, while action-level allocation maintains more stable KL dynamics.

while strictly maintaining outcome-driven optimization. Appendix D.2 presents further matched comparisons isolating loss-level distillation from advantage redistribution.

4.6 Hyperparameter Analysis

Redistribution strength. We evaluate the allowable weight deviation bound $\epsilon_w \in \{0.2, 0.4, 0.6, 0.8\}$ in Table 5. The default configuration ($\epsilon_w = 0.4$) achieves optimal task success (86.2%), whereas aggressive bounds induce noticeable performance drops. Mean-one normalization preserves the trajectory's average advantage for every tested value of ϵ_w. Increasing ϵ_w permits larger changes to individual decision weights, while decreasing it keeps those weights closer to one. The best result at $\epsilon_w = 0.4$ favors a moderate amount of redistribution within the tested range.

Granular boundary vs. Token noise. As shown in Table 5, action-level allocation achieves 86.2% success, exceeding the token-level variant's 79.7% by 6.5 percentage points. All four action-level settings outperform the token-level reference at $\epsilon_w = 0.4$, by at least 3.4 percentage points. Figure 4 further shows smoother reward improvement and fewer pronounced KL spikes. These observations support organizing credit around executable actions: TASPO aggregates PI-induced likelihood shifts over action tokens and applies a shared coefficient to the corresponding response, keeping the credit weights consistent within each action.

Table 5: **Effect of credit allocation settings.** Macro-averaged success on ALFWorld after 150 updates with Qwen2.5-3B-Instruct. We fix $\tau = 0.5$ and $c_\Delta = 2.0$.

Allocation unit	ϵ_w	ALF Avg. (%)
Action-level	0.2	83.1
Action-level	**0.4**	**86.2**
Action-level	0.6	84.2
Action-level	0.8	83.5
Token-level	0.4	79.7

5 Conclusion

In this paper, we presented TASPO, a framework that reframes the integration of process supervision and outcome feedback as a constrained credit-redistribution problem. Rather than optimizing process preferences as a competing objective, TASPO aligns privileged guidance from verified successful siblings with the target interaction history, aggregates likelihood shifts over executable actions, and maps them to positive, bounded, mean-one weights. This mechanism refines intermediate credit while strictly preserving the outcome advantage's polarity and mean advantage across turns, requiring no extra environment rollouts. Across nine benchmark–backbone configurations, TASPO consistently outperforms GRPO while stabilizing optimization dynamics. Our findings show that process guidance can improve multi-turn agent training by determining how outcome-derived reinforcement is distributed across decisions. This offers a promising perspective on integrating process supervision with outcome feedback through constrained credit redistribution.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Dingwei Chen, Zefang Zong, Zhipeng Ma, Leo Luo, Yang Li, Chengming Li, Peng Chen, and Jie Jiang. A^2TGPO: Agentic turn-group policy optimization with adaptive turn-level clipping. *arXiv preprint arXiv:2605.06200*, 2026. URL https://arxiv.org/abs/2605.06200.

Sanjiban Choudhury. Process reward models for llm agents: Practical framework and directions, 2025. URL https://arxiv.org/abs/2502.10325.

Tianyu Ding, Jianhong Xin, and Juan Pablo De la Cruz Weinstein. Keep policy gradient in charge: Sibling-guided credit distillation for long-horizon tool-use agents. *arXiv preprint arXiv:2606.12634*, 2026. URL https://arxiv.org/abs/2606.12634.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. ReTool: Reinforcement learning for strategic tool use in LLMs. *arXiv preprint arXiv:2504.11536*, 2025a. URL https://arxiv.org/abs/2504.11536.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/420c9f777c0b4f78d515e53cf74d58b2-Abstract-Conference.html.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026. URL https://arxiv.org/abs/2603.25562.

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. *arXiv preprint arXiv:2503.09516*, 2025. URL https://arxiv.org/abs/2503.09516.

Gengsheng Li, Mao Zheng, Mingyang Song, Ruiqi Liu, Tianyu Yang, Jie Sun, Qiyong Zhong, Haiyun Guo, Junfeng Fang, Dan Zhang, and Jinqiao Wang. On-policy distillation with curriculum turn-level guidance for multi-turn agents. *arXiv preprint arXiv:2606.15912*, 2026a. URL https://arxiv.org/abs/2606.15912.

Jian Li, Dongsheng Chen, Zhenhua Xu, Yizhang Jin, Jiafu Wu, Chengjie Wang, Xiaotong Yuan, and Yabiao Wang. Improving search agent with one line of code. *arXiv preprint arXiv:2603.10069*, 2026b. URL https://arxiv.org/abs/2603.10069.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, and Ning Ding. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026c. URL https://arxiv.org/abs/2604.13016.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2605.15155*, 2026. URL https://arxiv.org/abs/2605.15155.

Zibin Meng and Kani Chen. CRAFT: Counterfactual credit assignment from free sibling rollouts for self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2606.29476*, 2026. URL https://arxiv.org/abs/2606.29476.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-SD: Skill-conditioned self-distillation for multi-turn LLM agents. *arXiv preprint arXiv:2604.10674*, 2026a. URL https://arxiv.org/abs/2604.10674.

Xucong Wang, Ziyu Ma, Yong Wang, Yuxiang Ji, Shidong Yang, Guanhua Chen, Pengkun Wang, and Xiangxiang Chu. APPO: Agentic procedural policy optimization. *arXiv preprint arXiv:2606.12384*, 2026b. URL https://arxiv.org/abs/2606.12384.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Fei-Fei Li, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025. URL https://arxiv.org/abs/2504.20073.

Jinyang Wu, Shuo Yang, Zhengxi Lu, Fan Zhang, Yuhao Shen, Lang Feng, Haoran Luo, Zheng Lian, Shuai Zhang, Zhengqi Wen, and Jianhua Tao. SEED: Self-evolving on-policy distillation for agentic reinforcement learning. *arXiv preprint arXiv:2607.14777*, 2026. URL https://arxiv.org/abs/2607.14777.

Yan Xie, Sijie Zhu, Tiansheng Wen, Bo Chen, and Yifei Wang. On the position bias of on-policy distillation. *arXiv preprint arXiv:2606.22600*, 2026. URL https://arxiv.org/abs/2606.22600.

Xingrun Xing, Haoqing Wang, Boyan Gao, Ziheng Li, and Yehui Tang. Trust region on-policy distillation. *arXiv preprint arXiv:2606.01249*, 2026. URL https://arxiv.org/abs/2606.01249.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. TIP: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026. URL https://arxiv.org/abs/2604.14084.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled RLVR. *arXiv preprint arXiv:2604.03128*, 2026a. URL https://arxiv.org/abs/2604.03128.

Kaibing Yang, Guangfeng Cai, Shengtian Yang, Shuo He, Yu Li, Mengyi Liu, Pengwei Chen, Jun Xu, and Lei Feng. Progress-conditioned group policy optimization for long-horizon agentic tasks. *arXiv preprint arXiv:2607.22724*, 2026b. URL https://arxiv.org/abs/2607.22724.

Shuo Yang, Jinyang Wu, Zhengxi Lu, Yuhao Shen, Fan Zhang, Lang Feng, Shuai Zhang, Haoran Luo, Zheng Lian, Zhengqi Wen, and Jianhua Tao. OPID: On-policy skill distillation for agentic reinforcement learning. *arXiv preprint arXiv:2606.26790*, 2026c. URL https://arxiv.org/abs/2606.26790.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20744–20757, 2022. URL https://arxiv.org/abs/2207.01206.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026. URL https://arxiv.org/abs/2602.12275.

Yanfei Zhang, Xu Lin, and Chenglin Wu. StepOPSD: Step-aware online preference distillation for agent reinforcement learning. *arXiv preprint arXiv:2605.27140*, 2026. URL https://arxiv.org/abs/2605.27140.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026. URL https://arxiv.org/abs/2601.18734.

Zefang Zong, Dingwei Chen, Yang Li, Qi Yi, Bo Zhou, Chengming Li, Bo Qian, Peng Chen, and Jie Jiang. AT^{2}PO: Agentic turn-based policy optimization via tree search. *arXiv preprint arXiv:2601.04767*, 2026. URL https://arxiv.org/abs/2601.04767.

Algorithm 1 One TASPO Training Iteration

Require: Policy π_θ, frozen analyzer, task batch $\mathcal{B}$, group size K, token-shift clipping threshold $c_\Delta > 0$, temperature $\tau > 0$, deviation budget $0 \le \epsilon_w < 1$, and $\eta = \epsilon_w/2$

Freeze the rollout snapshot: $\theta_{\text{old}} \leftarrow \theta$
for each task $x \in \mathcal{B}$ **do**
  Sample K trajectories $\mathcal{G}_x = \{\tau_i\}_{i=1}^K$ with $\pi_{\theta_{\text{old}}}$; compute rewards R_i and group-relative advantages A_i
  Extract and validate source evidence $\mathcal{E}_x$ from verified successes $\mathcal{S}_x \subseteq \mathcal{G}_x$; use $\mathcal{E}_x = \varnothing$ if no valid evidence is available
  for each target trajectory $\tau_i \in \mathcal{G}_x$ **do**
    Initialize $w_{it} \leftarrow 1$ for all $t = 1, \ldots, T_i$
    Remove evidence supported only by τ_i, obtaining $\mathcal{E}_{x,-i}$
    Align $\mathcal{E}_{x,-i}$ with task x and the completed target history $\mathcal{H}_i$, excluding explicit outcome metadata; validate and compose P_i
    Use $P_i = \varnothing$ if no validated guidance remains
    if $P_i \neq \varnothing$ **and** $A_i \neq 0$ **then**
      Construct paired inputs, parse action spans, and rescore the sampled responses with $\pi_{\theta_{\text{old}}}$
      Collect scorable turns $\mathcal{I}_i$ using Appendix A.3
      if $|\mathcal{I}_i| \ge 2$ **then**
        Compute d_{it} for $t \in \mathcal{I}_i$ using Eq. equation A.1
        $\widehat{d}_{it} \leftarrow d_{it} - \frac{1}{|\mathcal{I}_i|}\sum_{u \in \mathcal{I}_i} d_{iu}$ for $t \in \mathcal{I}_i$
        $q_{it} \leftarrow \tanh\Big(\operatorname{sign}(A_i)\widehat{d}_{it}/\tau\Big)$ for $t \in \mathcal{I}_i$
        $\bar{q}_i \leftarrow \frac{1}{|\mathcal{I}_i|}\sum_{u \in \mathcal{I}_i} q_{iu}$
        $w_{it} \leftarrow 1 + \eta(q_{it} - \bar{q}_i)$ for $t \in \mathcal{I}_i$
      end if
    end if
    Assign detached $\widetilde{A}_{it} = A_i w_{it}$ to all generated response tokens $\mathcal{R}_{it}$ in each turn
  end for
end for
Update θ under the original contexts using Eq. equation 7

A Method and Implementation Details

This appendix specifies how source evidence is converted into turn-level credit weights. We describe one training iteration, the analyzer protocol, and the scoring and loss implementation, including the conditions under which TASPO falls back to uniform outcome credit.

A.1 Training Algorithm

Algorithm 1 summarizes one TASPO training iteration. At the start of each iteration, we snapshot the current policy for rollout collection and paired scoring. The analyzer and rollout snapshot remain frozen throughout the iteration; only π_θ is updated. Privileged information (PI) is used only for retrospective scoring, whereas policy optimization evaluates the sampled responses under their original contexts. Test-time generation does not use PI.

If verified source evidence is unavailable, no guidance survives alignment, $A_i = 0$, or fewer than two turns are scorable, the corresponding weights remain one. Under the same loss normalization as Eq. equation 7, this recovers the uniform-credit GRPO reference. In particular, TASPO introduces no additional process guidance when successful source evidence is unavailable.

A.2 Analyzer Protocol and Prompts

Source support and target applicability. Guidance extracted from a successful trajectory may depend on conditions that do not hold along another trajectory. We therefore separate source evidence extraction from target-path alignment. Extraction identifies source-supported task requirements, state-conditioned progress rules, constraints, and alternative strategies. For target τ_i, evidence supported only by that target is removed, so each remaining candidate retains support from another verified success. Alignment then checks each candidate against task x and the completed target history $\mathcal{H}_i$. Retained statements cite both source evidence and target support before being assembled into the

shared privileged context P_i. These checks provide traceability and an inspectable basis for guidance; citation matching alone does not establish causal relevance.

Input traces and outcome visibility. Both stages receive compact interaction traces as JSON user messages. Extraction receives verified successful trajectories, whereas alignment receives the target task, all recorded target steps, and the evidence items allowed for that target. The target payload excludes explicit outcome metadata, including `verified_success`, `episode_reward`, and precomputed advantages. Recorded observations are left unchanged and may themselves reveal task completion. Thus, the exclusion applies to explicit outcome labels rather than to all information correlated with the outcome. Prompts 1 and 2 require grounding in the supplied interactions and abstention when the evidence is insufficient.

Prompt 1: Source evidence extraction

You are a conservative training-time evidence analyst for a language-model agent. You receive only environment-verified successful trajectories for one task. Abstract reusable guidance without inventing states, observations, actions, or outcomes.
Rules:

1. Extract task requirements, state-conditioned progress rules, constraints, and valid alternative paths. Do not turn one trajectory into a single mandatory plan.
2. Every item must cite exact evidence from at least one supplied successful trajectory: trajectory id, step, an environment-accepted action quote, and a pre- or post-observation quote. Never use an invalid action as positive source evidence.
3. A statement may be more abstract than its quote, but it must be entailed by the cited action and environment observation.
4. Do not assign action scores, advantages, or numeric weights. Do not diagnose trajectories that are not supplied.
5. If the evidence is insufficient, return fewer items or an empty list.

Return one JSON object only:

```
{ "evidence": [{
      "evidence_id": "E1",
      "kind": "requirement|progress|constraint|alternative",
      "statement": "concise natural-language guidance",
      "sources": [{
          "traj_uid": "...", "step": 0,
          "pre_quote": "exact short quote or empty string",
          "action_quote": "exact action quote",
          "post_quote": "exact short quote or empty string"
        }]
}] }
```

Prompt 2: Trajectory-level alignment

Align source-supported guidance to each target's recorded task, actions, and observations. Explicit terminal reward and success labels are not supplied. Recorded observations are immutable.
Rules:

1. For each target, use only its allowed_evidence items and ids. These candidates have supporting evidence from at least one successful trajectory other than the target.
2. Inspect all supplied steps of the realized target trajectory. Check each item's applicability against the task and recorded states and actions. Source-supported guidance may still be inapplicable to this target path.
3. Base applicability on specific interaction evidence. Do not judge an action solely from an apparent overall outcome or output inferred terminal labels.
4. Every retained statement must cite allowed source evidence ids and exact target support from the task, action, pre-observation, or post-observation.
5. You may reformulate guidance as a target-specific warning or requirement when supported jointly by the cited source and target evidence. Never fabricate observations or counterfactual rollouts. Produce one guidance set for the target trajectory.

Prompt 2: Trajectory-level alignment (continued)

6. Do not output action scores, advantages, importance weights, or numeric updates. If the evidence does not support alignment, return empty guidance and an abstention reason.

Return one JSON object only:

```
{ "targets": [{
      "traj_uid": "...",
      "guidance": [{
          "statement": "concise guidance for the shared PI",
          "evidence_ids": ["E1"],
          "target_support": [{
              "step": 2,
              "field": "task|pre|action|post",
              "quote": "exact short quote"
            }]
        }],
      "abstain_reason": "empty when guidance is retained"
}] }
```

Output validation. Prompt outputs must follow the required JSON structure. Source items require unique nonempty identifiers, nonempty statements of at most 800 characters, and references to valid actions in verified successful trajectories. Each source reference must match its action quote and at least one pre- or post-observation quote. Alignment requires a known target, allowed source evidence identifiers, a nonempty statement of at most 800 characters, and a supported target quote. Quotes are checked by whitespace-normalized substring matching against the original records. These checks establish reference validity and traceability; semantic entailment and applicability remain analyzer judgments. Rejected items are discarded, and a target with no validated guidance receives $P_i = \varnothing$.

Interaction trace compaction. Compaction retains an entry for every recorded step while shortening overlong fields. After whitespace normalization, each overlong field retains its beginning and end, separated by `...`. The character budget C is distributed across the task and interaction steps, with task and action caps of 1,200 and 320 characters and a minimum observation budget of 40 characters. Field minima and JSON overhead make C an approximate request budget. Retaining every step does not guarantee that every detail survives field truncation.

PI composition and budgeting. Accepted statements among the first G candidates are concatenated as bullets and shared across all turns of trajectory τ_i. We denote the PI token budget by B_{PI}, separately from the trajectory batch size B. If the markers `[Privileged Guidance]` and `[End Privileged Guidance]` occupy $B_{\mathrm{wrapper}} < B_{\mathrm{PI}}$ tokens, they enclose the first $B_{\mathrm{PI}} - B_{\mathrm{wrapper}}$ guidance tokens. Otherwise, the first B_{PI} guidance tokens are used without wrappers. The original scoring contexts and sampled responses are preserved. Truncation may cut a guidance item; validation applies to the original statement and therefore does not guarantee completeness of its retained prefix.

Default hyperparameters. We use $C = 12{,}000$, $G = 8$, $B_{\mathrm{PI}} = 256$, and an analyzer output limit of 4,096 tokens. The scoring defaults are $c_\Delta = 2.0$, $\tau = 0.5$, and $\epsilon_w = 0.4$, giving $\eta = \epsilon_w/2 = 0.2$.

A.3 Scoring and Loss Implementation

Scoring evidence and credit recipients. The action mask $\mathcal{M}_{it}$ selects the executable span over which likelihood shifts are aggregated, whereas $\mathcal{R}_{it}$ contains all generated response tokens that receive the resulting turn-level coefficient. Scoring preserves the sampled response and its autoregressive prefix, including any preceding reasoning. Thus, action-only scoring restricts score aggregation rather than removing reasoning tokens from the conditioning sequence.

Action span identification. We locate the first complete `<action>`, `<search>`, or `<answer>` block using case-insensitive matching across lines. After decoding response tokens individually, we include tokens overlapping the matched span, including both tags. If no block is found through this procedure, the same pattern is applied to the recorded response text. The complete matched block is then tokenized, with or without one leading space, and located as an exact subsequence of the response tokens. If neither procedure locates a complete span, $\mathcal{M}_{it} = \varnothing$. This fallback still requires a complete tagged block; it does not infer executable spans from untagged text.

Paired scoring and valid turns. Both passes evaluate the same sampled response using the frozen rollout snapshot $\pi_{\theta_{\text{old}}}$, adding P_i only to the privileged context. A turn enters $\mathcal{I}_i$ when both scoring inputs are available, $\mathcal{M}_{it}$ is complete and nonempty, and every action-token log probability and corresponding shift Δ_{itk} in Eq. equation 3 is finite. Otherwise, the turn is excluded from scoring and retains unit weight. These checks avoid comparing partial-action scores with complete-action scores.

Clipped action support and centering. For each $t \in \mathcal{I}_i$, we compute

$$d_{it} = \frac{1}{|\mathcal{M}_{it}|} \sum_{k \in \mathcal{M}_{it}} \text{clip}(\Delta_{itk}, -c_\Delta, c_\Delta), \qquad c_\Delta > 0. \tag{A.1}$$

Clipping limits the contribution of individual token shifts before averaging over the complete action span; it does not discard outlier tokens. No small-magnitude filter is applied. Both centering operations use the same set $\mathcal{I}_i$: centering d_{it} before the $\tanh$ transformation produces relative support, while centering q_{it} in the weight construction produces zero-sum weight deviations.

Local and trajectory-level fallback. Unscorable turns retain $w_{it} = 1$. Empty PI, $A_i = 0$, or $|\mathcal{I}_i| < 2$ leaves all weights at one. Otherwise, redistribution occurs only within $\mathcal{I}_i$. Because the corrections sum to zero over this set and vanish elsewhere, the complete trajectory satisfies

$$\sum_{t=1}^{T_i} (w_{it} - 1) = 0, \qquad \frac{1}{T_i} \sum_{t=1}^{T_i} \widetilde{A}_{it} = A_i.$$

Appendix B.1 gives the corresponding derivation and bounds.

Response-level application and loss normalization. The detached advantage $\widetilde{A}_{it} = A_i w_{it}$ is applied to all generated tokens in $\mathcal{R}_{it}$, including reasoning tokens; environment observations and padding are masked out. When reasoning precedes an action, that action is generated conditionally on the reasoning prefix. Using one coefficient therefore treats the generated response as a turn-level training unit without independently assessing the correctness or causal contribution of its reasoning tokens.

For a row-averaged implementation of Eq. equation 7, multiply each real turn's token-averaged loss by

$$\frac{N_{\text{row}}}{N_{\text{traj}} T_i}.$$

Here, N_{row} is the number of rows in the reduction batch, including padded rows that contribute zero, and N_{traj} is the number of real trajectories in that batch. This factor implements the nested trajectory and turn averaging when the outer reduction is an arithmetic mean over all N_{row} rows. Other row-reduction conventions require the corresponding normalization adjustment. This balancing factor is separate from the mean-one credit weights and remains active during unit-weight fallback.

B Derivation and Properties

B.1 Weight Derivation and Credit Preservation

For a fixed trajectory with $A_i \neq 0$ and $n = |\mathcal{I}_i| \geq 2$, let $q_{it} = \tanh(\text{sign}(A_i)\widehat{d}_{it}/\tau)$ and $\eta = \epsilon_w/2$. The weight rule admits the following constrained allocation interpretation:

$$\begin{aligned} \max_{\{w_{it}\}_{t \in \mathcal{I}_i}} \quad & \eta \sum_{t \in \mathcal{I}_i} q_{it}(w_{it} - 1) - \frac{1}{2} \sum_{t \in \mathcal{I}_i} (w_{it} - 1)^2 \\ \text{s.t.} \quad & \sum_{t \in \mathcal{I}_i} (w_{it} - 1) = 0, \qquad |w_{it} - 1| \leq \epsilon_w. \end{aligned} \tag{B.1}$$

The linear term favors larger weights for larger q_{it}, while the quadratic term penalizes departures from uniform credit.

Derivation. With only the equality constraint, stationarity gives $w_{it} - 1 = \eta q_{it} - \lambda$. Summing over $\mathcal{I}_i$ gives $\lambda = \eta \bar{q}_i$, where $\bar{q}_i = n^{-1} \sum_{t \in \mathcal{I}_i} q_{it}$. Therefore,

$$w_{it}^* = 1 + \eta(q_{it} - \bar{q}_i). \tag{B.2}$$

Since $q_{it}, \bar{q}_i \in [-1, 1]$, $|w^*_{it} - 1| \le 2\eta = \epsilon_w$. The solution already satisfies the box constraints. Strict concavity therefore makes it the unique optimizer of Eq. equation B.1, recovering Eq. equation 6 without an additional optimization step.

Preserved quantities. Use Eq. equation B.2 on scorable turns and unit weights elsewhere. The centered corrections sum to zero within $\mathcal{I}_i$, so the complete trajectory has $\sum_{t=1}^{T_i} w_{it} = n + (T_i - n) = T_i$. For $0 \le \epsilon_w < 1$,

$$1 - \epsilon_w \le w_{it} \le 1 + \epsilon_w, \qquad w_{it} > 0, \qquad \frac{1}{T_i} \sum_{t=1}^{T_i} w_{it} = 1. \tag{B.3}$$

Consequently,

$$\begin{aligned} &\operatorname{sign}(\widetilde{A}_{it}) = \operatorname{sign}(A_i), \quad \frac{1}{T_i} \sum_{t=1}^{T_i} \widetilde{A}_{it} = A_i, \\ &|\widetilde{A}_{it} - A_i| \le \epsilon_w |A_i|. \end{aligned} \tag{B.4}$$

The unit-weight fallback satisfies the same properties. In particular, an action on a negative-advantage trajectory retains negative credit even when guidance supports it; redistribution can attenuate that credit without reversing its sign. These are scalar advantage guarantees. The parameter-gradient direction also depends on the individual token gradients and can change under reweighting.

B.2 Clipped Scores and Relative Support

Bounded token influence. Consider a fixed scorable action with L tokens and let $f_c(z) = \operatorname{clip}(z, -c, c)$, with $c = c_\Delta$. Since $|f_c(z)| \le c$,

$$d = \frac{1}{L} \sum_{k=1}^{L} f_c(\Delta_k), \qquad |d| \le c. \tag{B.5}$$

If only token j changes from Δ_j to Δ'_j, clipping is 1-Lipschitz and has range $[-c, c]$, giving

$$|d' - d| \le \frac{1}{L} \min\{|\Delta'_j - \Delta_j|,\, 2c\}. \tag{B.6}$$

Thus, one token changes the action mean by at most $2c/L$, assuming a fixed action span and unchanged scoring validity. The bound makes no assumption about which shifts reflect task value.

Shared-shift invariance. For fixed $\mathcal{I}_i$, add a constant b_i to every aggregated action score: $d'_{it} = d_{it} + b_i$. Then $d'_{it} - \bar{d}'_i = (d_{it} + b_i) - (\bar{d}_i + b_i) = \widehat{d}_{it}$, so q_{it} and w_{it} remain unchanged. This applies to aggregated scores; shared offsets before token clipping need not cancel because clipping is nonlinear. The two centerings establish relative support and mean-one weights, respectively.

B.3 Compatibility with the Policy Objective

Let r be the current-to-rollout policy probability ratio for a sampled token under its original context. Write the clipped surrogate to be maximized as $s_{\mathrm{clip}}(r, A) = \min\{rA, \operatorname{clip}(r, 1-\epsilon_{\mathrm{PPO}}, 1+\epsilon_{\mathrm{PPO}})A\}$. For $A < 0$, define the dual-clipped surrogate $s_{\mathrm{dc}}(r, A) = \max\{s_{\mathrm{clip}}(r, A), c_{\mathrm{dc}}A\}$ with $c_{\mathrm{dc}} > 1$; otherwise it equals $s_{\mathrm{clip}}(r, A)$. For a fixed detached $w > 0$,

$$s_{\mathrm{clip}}(r, wA) = w\, s_{\mathrm{clip}}(r, A), \qquad s_{\mathrm{dc}}(r, wA) = w\, s_{\mathrm{dc}}(r, A). \tag{B.7}$$

Positive homogeneity of minimum and maximum gives both identities; the dual-clipped sign condition is also preserved. They apply equally to the negative-surrogate policy loss. At fixed r, the clipping branch is unchanged. KL and entropy retain their separate coefficients.

B.4 Bounded Gradient Perturbation

Mean-one redistribution restricts the gradient change to differences among turns of the same trajectory. Fix a rollout batch and evaluate both policy losses at the same differentiable parameter value θ. The rollout policy, sampled responses, masks, A_i, and w_{it} are held fixed during differentiation. We

analyze the advantage-dependent policy term using exactly the trajectory and turn averaging in Eq. equation 7; its uniform-credit reference sets $w_{it} = 1$ while retaining this averaging.

Let the original turn-level gradient contribution and its trajectory mean be

$$\mathbf{v}_{it} = \frac{1}{|\mathcal{R}_{it}|} \sum_{k \in \mathcal{R}_{it}} \nabla_\theta \ell_{itk}^{\mathrm{GRPO}}(\theta; A_i), \qquad \overline{\mathbf{v}}_i = \frac{1}{T_i} \sum_{t=1}^{T_i} \mathbf{v}_{it}. \tag{B.8}$$

The positive homogeneity established in Appendix B.3 gives

$$\mathbf{g}_{\mathrm{U}} = \frac{1}{B} \sum_{i=1}^{B} \overline{\mathbf{v}}_i, \qquad \mathbf{g}_{\mathrm{T}} = \frac{1}{B} \sum_{i=1}^{B} \frac{1}{T_i} \sum_{t=1}^{T_i} w_{it} \mathbf{v}_{it}. \tag{B.9}$$

Define $\delta_{it} = w_{it} - 1$ and the within-trajectory dispersions

$$\sigma_{w,i}^2 = \frac{1}{T_i} \sum_{t=1}^{T_i} \delta_{it}^2, \qquad S_i^2 = \frac{1}{T_i} \sum_{t=1}^{T_i} \|\mathbf{v}_{it} - \overline{\mathbf{v}}_i\|_2^2. \tag{B.10}$$

Proposition (Bounded gradient perturbation). For the TASPO weights, including unit-weight fallbacks, the perturbation $\mathbf{e} = \mathbf{g}_{\mathrm{T}} - \mathbf{g}_{\mathrm{U}}$ satisfies

$$\mathbf{e} = \frac{1}{B} \sum_{i=1}^{B} \frac{1}{T_i} \sum_{t=1}^{T_i} \delta_{it} (\mathbf{v}_{it} - \overline{\mathbf{v}}_i), \tag{B.11}$$

$$\|\mathbf{e}\|_2 \le \frac{1}{B} \sum_{i=1}^{B} \sigma_{w,i} S_i \le \frac{\epsilon_w}{2B} \sum_{i=1}^{B} S_i =: \Gamma. \tag{B.12}$$

Proof. Mean-one weights give $\sum_t \delta_{it} = 0$, so subtracting $\overline{\mathbf{v}}_i$ leaves each trajectory's perturbation unchanged. The triangle and Cauchy–Schwarz inequalities then give the first inequality in Eq. equation B.12. For an active trajectory, let $n_i = |\mathcal{I}_i|$. Using $\delta_{it} = \eta(q_{it} - \bar{q}_i)$ on $\mathcal{I}_i$, zero corrections elsewhere, and $\eta = \epsilon_w/2$,

$$\sigma_{w,i}^2 = \eta^2 \frac{n_i}{T_i} \left(\frac{1}{n_i} \sum_{t \in \mathcal{I}_i} q_{it}^2 - \bar{q}_i^2 \right) \le \eta^2 \frac{n_i}{T_i} \le \eta^2, \tag{B.13}$$

because $|q_{it}| \le 1$. A trajectory-level fallback has $\sigma_{w,i} = 0$. This proves the second inequality. □

Equation equation B.11 explains the role of mean preservation: a gradient component shared by all turns of a trajectory cancels from the correction. In particular, equal turn gradients give zero perturbation, while Eq. equation B.12 controls its magnitude when turn gradients differ.

Corollary (Conditional directional alignment). If $\Gamma < \|\mathbf{g}_{\mathrm{U}}\|_2$, then

$$\langle \mathbf{g}_{\mathrm{T}}, \mathbf{g}_{\mathrm{U}} \rangle > 0, \qquad \angle(\mathbf{g}_{\mathrm{T}}, \mathbf{g}_{\mathrm{U}}) \le \arcsin\left(\frac{\Gamma}{\|\mathbf{g}_{\mathrm{U}}\|_2} \right). \tag{B.14}$$

Indeed, $\langle \mathbf{g}_{\mathrm{T}}, \mathbf{g}_{\mathrm{U}} \rangle \ge \|\mathbf{g}_{\mathrm{U}}\|_2 (\|\mathbf{g}_{\mathrm{U}}\|_2 - \Gamma) > 0$. Writing α for the resulting acute angle, the distance from $\mathbf{g}_{\mathrm{U}}$ to the ray through $\mathbf{g}_{\mathrm{T}}$ is $\|\mathbf{g}_{\mathrm{U}}\|_2 \sin\alpha$ and is at most $\|\mathbf{e}\|_2 \le \Gamma$, proving the angle bound.

These statements concern full-batch raw policy gradients before gradient clipping and optimizer transformations. Identical KL and entropy terms cancel from the gradient difference; directional alignment with the regularized gradient instead requires the same condition using that regularized reference. A minibatch containing only part of a trajectory need not preserve the zero-sum correction.

B.5 Comparison with Joint Optimization: Bias and Variance

Direct joint optimization adds a separate distillation gradient, $\mathbf{g}_{\mathrm{J}} = \mathbf{g}_{\mathrm{U}} + \lambda \mathbf{d}$, where $\lambda \ge 0$ is fixed and $\mathbf{d}$ is evaluated using the same policy parameters, sampled batch, and PI construction. TASPO instead gives $\mathbf{g}_{\mathrm{T}} = \mathbf{g}_{\mathrm{U}} + \mathbf{e}$, with $\mathbf{e}$ satisfying Eq. equation B.12. Both comparisons retain the same base-loss normalization.

Hold the current and rollout-policy parameters fixed and take expectations over rollout batches and any analyzer randomness. Assume finite second moments for the gradients and Γ. For a vector-valued estimator X, write $V(X) = \mathbb{E}\|X - \mathbb{E}X\|_2^2$ and $C(X, Y) = \mathbb{E}\langle X - \mathbb{E}X, Y - \mathbb{E}Y\rangle$. Relative to a fixed target gradient $\mathbf{g}_\star$ defined independently of the sampled batch, let $\mathbf{b}_m = \mathbb{E}\mathbf{g}_m - \mathbf{g}_\star$. Direct expansion gives

$$\mathbf{b}_{\mathrm{T}} = \mathbf{b}_{\mathrm{U}} + \mathbb{E}\mathbf{e}, \qquad \mathbf{b}_{\mathrm{J}} = \mathbf{b}_{\mathrm{U}} + \lambda\mathbb{E}\mathbf{d}. \tag{B.15}$$

$$V(\mathbf{g}_{\mathrm{T}}) = V(\mathbf{g}_{\mathrm{U}}) + V(\mathbf{e}) + 2C(\mathbf{g}_{\mathrm{U}}, \mathbf{e}), \tag{B.16}$$

$$V(\mathbf{g}_{\mathrm{J}}) = V(\mathbf{g}_{\mathrm{U}}) + \lambda^2 V(\mathbf{d}) + 2\lambda C(\mathbf{g}_{\mathrm{U}}, \mathbf{d}). \tag{B.17}$$

No independence between weights and sampled gradients is used. For TASPO, the batchwise bound also yields

$$\|\mathbf{b}_{\mathrm{T}} - \mathbf{b}_{\mathrm{U}}\|_2 \le \mathbb{E}\Gamma, \qquad V(\mathbf{e}) \le \mathbb{E}\|\mathbf{e}\|_2^2 \le \mathbb{E}\Gamma^2. \tag{B.18}$$

The redistribution budget therefore bounds the additional bias and perturbation variance. Whether total variance decreases depends on the covariance term; for example, TASPO reduces $V(\mathbf{g}_{\mathrm{U}})$ precisely when $2C(\mathbf{g}_{\mathrm{U}}, \mathbf{e}) < -V(\mathbf{e})$. Joint optimization can also control its perturbation through λ and the scale of $\mathbf{d}$; comparative performance depends on these moments and the quality of process guidance.

Choosing $\mathbf{g}_\star = \mathbb{E}\mathbf{g}_{\mathrm{U}}$ measures deviation from the uniform-credit surrogate. Interpreting bias relative to the expected-return gradient additionally requires specifying that objective and accounting for the reference estimator's own bias. The results above establish finite-batch perturbation control. Convergence rates for the changing on-policy training objective require further assumptions on sampling, guidance, and optimizer dynamics.

C Experimental Details

This appendix specifies the evaluation protocols, training configuration, and sensitivity to the training-time analyzer. It complements Appendix A by defining the benchmark splits, metric aggregation, and default settings used in the experiments.

C.1 Evaluation Protocols

Table C.1 summarizes the training data and evaluation sets. TASPO and the controlled reruns use the prescribed final checkpoint after 150 policy updates, with results averaged over three training seeds. Held-out tasks are excluded from training and checkpoint selection. Results taken from prior work are reported separately from controlled reruns.

Table C.1: Training data, evaluation sets, and reported metrics. ALFWorld and Search-QA additionally report the macro-averages defined below. Performance metrics are presented in percent.

Domain	Training data	Evaluation	Metrics
ALFWorld	Training split	140 seen tasks; 134 unseen tasks	Task-family success rates
WebShop	Training split	128 fixed test tasks	Normalized task score; success rate
Search-QA	NQ and HotpotQA	Seven datasets listed below	Exact-match accuracy

ALFWorld. We evaluate the six canonical task families: Pick, Look, Clean, Heat, Cool, and Pick2. The primary comparison uses the 140-task seen split, with matched reruns also conducted on the 134-task unseen split. The reported ALF Avg. is the unweighted mean of the six task-family success rates rather than a pooled success rate across individual tasks.

WebShop. We evaluate on 128 fixed test tasks and report both normalized task score and task success rate. The two metrics are reported separately as Score and Succ., respectively, following the main results table.

Search-QA. Policies are trained on Natural Questions (NQ) and HotpotQA. Evaluation covers NQ, TriviaQA (Triv), PopQA (Pop), HotpotQA (Hotp), 2WikiMultihopQA (2Wk), MuSiQue (MuS), and Bamboogle (Bam). The evaluation suite therefore contains held-out examples from the training

datasets as well as datasets not used for policy training. We report exact-match accuracy for each dataset and the unweighted mean across all seven datasets as Search Avg.

Cross-benchmark aggregation. The reported average improvement over GRPO across the nine benchmark–backbone settings uses ALFWorld macro-averaged success, Search-QA macro-averaged accuracy, and WebShop success rate. Each of the nine settings receives equal weight, and improvements are reported in percentage points.

C.2 Training and TASPO Configuration

We use Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, and Qwen3-1.7B-Instruct. Controlled analyses use Qwen2.5-3B-Instruct unless otherwise specified. For controlled comparisons, TASPO and the rerun baselines use the same backbone, scheduled rollout budget, number of policy updates, and evaluation tasks.

Policy optimization and rollout collection. Table C.2 reports the shared and domain-specific training settings. Each update samples $K = 8$ trajectories per task, with rollout temperature and top-p both set to 1.0. The group-relative advantage is computed from terminal rewards. The low-variance KL estimator listed in the table is used for KL regularization rather than for estimating this advantage. TASPO and the controlled reruns use the same trajectory and turn normalization in Eq. equation 7. The prompt and response limits in the table describe the training and rollout configuration.

Table C.2: Policy optimization and rollout configuration. Settings in the lower block are shared across all three domains. Rollout sampling temperature is distinct from the TASPO score temperature τ.

Setting	ALFWorld	WebShop	Search-QA
Tasks per update	16	16	128
Maximum interaction turns	50	15	4
Maximum prompt tokens	2,048	4,096	4,096
KL coefficient	0.01	0.01	0.001
Warmup ratio	0	0	0.1
PPO minibatch size	256	64	256
Microbatch size per GPU	16	8	16
Optimizer	AdamW		
Learning rate	10^{-6}		
Optimizer (β_1, β_2)	$(0.9, 0.999)$		
Weight decay / gradient clipping	0.01 / 1.0		
PPO clipping / dual-clip coefficient	0.2 / 3.0		
Policy epochs per batch	1		
Entropy coefficient	0.001		
KL estimator	Low-variance KL		
Rollouts per task K / policy updates	8 / 150		
Rollout temperature / top-p	1.0 / 1.0		
Maximum response tokens	512		
Number of training seeds	3		

Analyzer and redistribution settings. Table C.3 summarizes the default analyzer configuration, context budgets, and redistribution parameters. The analyzer performs source evidence extraction and target-path alignment, whereas the frozen rollout policy computes the paired likelihood shifts. The score temperature τ controls the transformation of relative action support and is distinct from both the rollout sampling temperature and the analyzer request temperature. With $\epsilon_w = 0.4$ and $\eta = \epsilon_w/2 = 0.2$, the resulting credit weights are bounded to $[0.6, 1.4]$. Appendix A.2 specifies how context compaction, candidate selection, and PI truncation use the listed budgets.

C.3 Sensitivity to the Training-Time Analyzer

The analyzer supplies textual guidance, whereas action scoring and credit redistribution are performed by the frozen rollout policy and the constrained weight rule. We evaluate sensitivity to analyzer choice by comparing DeepSeek-V4-Pro, GLM5.2, and Qwen3.5-35B-A3B on ALFWorld with Qwen2.5-

Table C.3: Default TASPO-specific settings. The allocation coefficient η is determined by ϵ_w rather than tuned independently. The token-shift threshold c_Δ clips shift magnitudes and does not filter out small shifts.

Setting	Value
Default analyzer	DeepSeek-V4-Pro
Analyzer temperature / reasoning mode	0 / disabled
Analyzer output limit / workers	4,096 tokens / 4
Maximum evidence items	12
PI-composition candidate limit G	8
Approximate input character budget C	12,000
PI token budget B_{PI}	256
Score temperature τ	0.5
Weight-deviation budget ϵ_w	0.4
Allocation coefficient $\eta = \epsilon_w/2$	0.2
Token-shift clipping threshold c_Δ	2.0

3B-Instruct, while keeping the downstream optimization configuration fixed. Table C.4 reports PI coverage together with final macro-averaged task-family success. PI coverage characterizes the availability of validated guidance rather than its semantic correctness.

Table C.4: Sensitivity to the training-time analyzer on ALFWorld. PI coverage and macro-averaged task-family success are reported with Qwen2.5-3B-Instruct under the same downstream optimization configuration. The default analyzer is highlighted.

Analyzer	PI coverage (%)	ALF Avg. (%)
DeepSeek-V4-Pro	72.4	**86.2**
GLM5.2	71.6	85.7
Qwen3.5-35B-A3B	72.1	86.0

Across the three evaluated analyzers, PI coverage ranges from 71.6% to 72.4%, while final ALFWorld macro-averaged success ranges from 85.7% to 86.2%. The corresponding spreads are 0.8 and 0.5 percentage points, respectively, indicating limited sensitivity to analyzer choice among these three models under this experimental setting. This comparison does not isolate the mechanism responsible for the similarity or establish independence from analyzer capabilities.

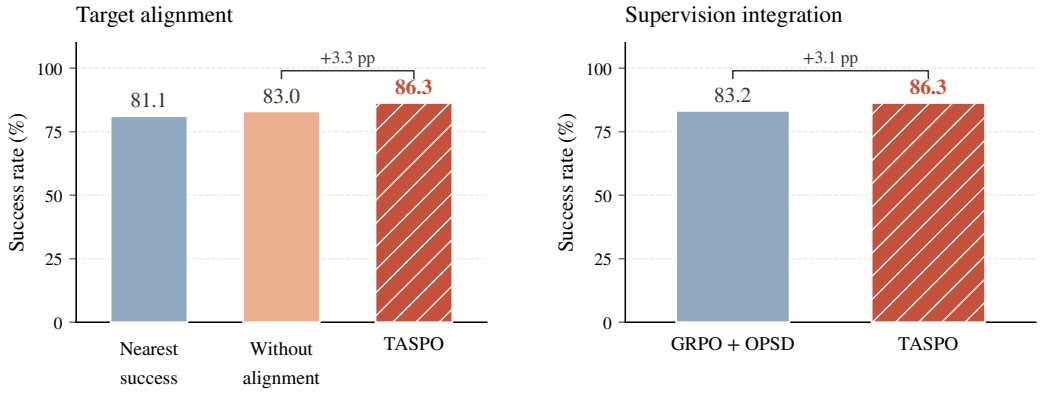
D Supplementary Experiments

The supplementary analyses examine four questions: whether source guidance benefits from target-path alignment, whether its effect depends on how it enters optimization, where action-derived credit should be applied, and whether action scores contain task-relevant ranking information. We then report computational costs, additional baseline comparisons, and training dynamics. The controlled ablations in Sections D.1–D.3 use ALFWorld with Qwen2.5-3B-Instruct, 150 policy updates, and three training seeds. The replay diagnostic and runtime profile use the setups specified in their respective sections.

D.1 Contribution of Target-Trajectory Alignment

Does source-supported guidance transfer to the target path? A rule extracted from a successful trajectory may depend on a precondition that is absent from the target interaction. We therefore test whether checking target applicability adds value beyond extracting conditional source guidance. The variant without alignment retains guidance extraction, source-evidence checks, exclusion of target-only evidence, and the same PI budget and credit-allocation rule.

Figure D.1(a) shows that source abstraction increases success from 81.1% with nearest-success reuse to 83.0%. Adding target alignment raises success to 86.3%. The additional 3.3 percentage points support checking applicability before using source guidance to allocate credit. The two stages serve



(a) Target alignment

(b) Supervision integration

Figure D.1: **Target alignment and supervision integration on ALFWorld.** Results use Qwen2.5-3B-Instruct. (a) Nearest-success reuse, source abstraction without target alignment, and the full TASPO pipeline. (b) GRPO+OPSD with aligned PI and TASPO under the same guidance-construction protocol. Each method constructs guidance from its own on-policy trajectories. Values are macro-averaged task-family success rates; differences are in percentage points.

complementary roles: abstraction expresses conditions supported by successful source interactions, whereas alignment checks their relevance to the recorded target path.

D.2 Integrating Aligned Process Supervision

Does the integration mechanism matter after matching guidance construction? We equip GRPO+OPSD with the same analyzer, extraction and alignment prompts, PI budget, and frozen-policy scoring procedure as TASPO. GRPO+OPSD incorporates the resulting supervision through an auxiliary distillation loss, whereas TASPO uses it to redistribute the original outcome advantage. Each method constructs PI from its own on-policy trajectories; the comparison therefore matches the construction protocol rather than the realized training data.

With this protocol, GRPO+OPSD reaches 83.2% success and TASPO reaches 86.3% in Figure D.1(b). The 3.1-point difference supports a benefit from redistribution beyond the choice of guidance-construction pipeline in this setting. The relevant design distinction is that TASPO uses guidance to adjust relative credit while preserving the sign and trajectory mean of the outcome-derived advantage.

D.3 Scope of Action-Derived Credit

How far should an action-derived coefficient extend? TASPO scores executable actions but applies the resulting coefficient to the full generated response associated with each action. When a response contains a reasoning prefix, the action is generated conditionally on that prefix. Applying one coefficient therefore treats the response as a turn-level training unit without assuming that every reasoning token is correct or causally necessary for the action.

We compare full-response allocation with an action-only variant that leaves reasoning tokens at the original advantage A_i. Both variants use the same action-scoring rule. They differ in the tokens receiving redistributed credit and in the centering required to preserve mean scalar credit under the turn-averaged objective. Directly applying the original TASPO weights only to action tokens would generally change the trajectory-average scalar credit under this objective.

We therefore account for the action-token fraction

$$\rho_{it} = \frac{|\mathcal{M}_{it}|}{|\mathcal{R}_{it}|}$$

and define

$$\bar{q}_i^{\mathrm{act}} = \frac{\sum_{t\in\mathcal{I}_i} \rho_{it} q_{it}}{\sum_{t\in\mathcal{I}_i} \rho_{it}}, \qquad w_{it}^{\mathrm{act}} = 1 + \eta\left(q_{it} - \bar{q}_i^{\mathrm{act}}\right), \qquad \eta = \epsilon_w/2. \tag{D.1}$$

These weights apply only to action tokens, with unit weights elsewhere and the same validity and fallback rules as TASPO. The identity

$$\sum_{t\in\mathcal{I}_i} \rho_{it}\left(w_{it}^{\mathrm{act}} - 1\right) = 0$$

preserves mean scalar credit under the common loss reduction. For $0 \le \epsilon_w < 1$, the weights remain positive and satisfy the same deviation bound as the full-response construction.

Table D.1: **Scope of action-derived credit on ALFWorld.** Results use Qwen2.5-3B-Instruct. Both variants score executable actions and preserve mean scalar credit under the turn-averaged loss. Action-only allocation uses action-fraction-weighted centering; TASPO reweights the full response.

Configuration	Reweighted tokens	ALF Avg. (%)
Action-only	Action tokens	84.0
TASPO	Full response	86.3

Table D.1 reports a 2.3-point advantage for full-response allocation. This result supports applying action-derived credit to the complete response under the reported normalization and per-token weight bounds. The comparison also changes the number of reweighted tokens and the effective redistribution strength, so it does not separately establish the correctness or causal contribution of the reasoning trace.

D.4 Credit Quality against Counterfactual Replay

Do action scores track replay-estimated utility? End-task improvements alone do not show whether the scores rank decisions meaningfully. We therefore compare pre-update action support with relative utilities estimated by branching the environment from recorded decision points. The diagnostic asks whether higher support is associated with more successful continuations under a fixed policy and a specified set of alternative actions.

Replay protocol. We sample 100 held-out evaluation trajectories across the six ALFWorld task families using a frozen Qwen2.5-3B-Instruct checkpoint. For each trajectory, we select four scorable decision points distributed across its temporal positions. At each point, we restore the recorded environment state and prompt history h_{it}. Alongside the executed action a_{it}, we sample three distinct valid alternatives $\{a_{it}^{(k)}\}_{k=1}^{3}$ from the rollout policy without PI. Each of the four actions starts 16 stochastic continuations under the same policy, without PI and within the remaining step budget.

Writing $\widehat{p}_{it}(a)$ for the empirical continuation success rate after action a, we define

$$\widehat{u}_{it} = \widehat{p}_{it}(a_{it}) - \frac{1}{3}\sum_{k=1}^{3} \widehat{p}_{it}\left(a_{it}^{(k)}\right). \tag{D.2}$$

A positive value indicates a higher estimated continuation success rate for the executed action than for the three sampled alternatives. This estimate is specific to the restored state, continuation policy, remaining horizon, and alternative-action set.

Scores and controls. We evaluate the ranking induced by d_{it} within each trajectory. Centering d_{it} to obtain $\widehat{d}_{it}$ leaves this ranking unchanged. The controls are: *policy confidence*, measured using action-token log probabilities under the original context; *raw successful-sibling PI*, which uses successful traces without target-path alignment; and *shuffled TASPO scores*, which permutes scores within each trajectory. The raw-sibling control is labeled "PI w/o alignment" in Figure D.2. It differs from the ablation in Section D.1, which retains source abstraction while removing target alignment.

We report trajectory-level Spearman rank correlation, pairwise ranking accuracy, and the replay-utility gap between the highest- and lowest-ranked selected actions in each trajectory. Pairwise accuracy

includes only action pairs whose estimated replay utilities have non-overlapping 95% bootstrap confidence intervals.

Table D.2: **Agreement with replay-estimated relative utility on ALFWorld.** Scores are evaluated within 100 held-out trajectories, with four selected decision points per trajectory. Pairwise accuracy is restricted to action pairs with non-overlapping 95% bootstrap intervals for their estimated replay utilities. The utility gap is the estimated success-probability difference, in percentage points, between the highest- and lowest-ranked selected actions.

Scoring signal	Spearman $\rho \uparrow$	Pairwise acc. (%)↑	Utility gap (pp)↑
Policy confidence	0.08	52.8	4.8
Raw successful-sibling PI	0.24	60.7	10.5
Shuffled TASPO scores	0.00	50.0	-0.2
TASPO (aligned PI)	**0.41**	**69.6**	**19.6**

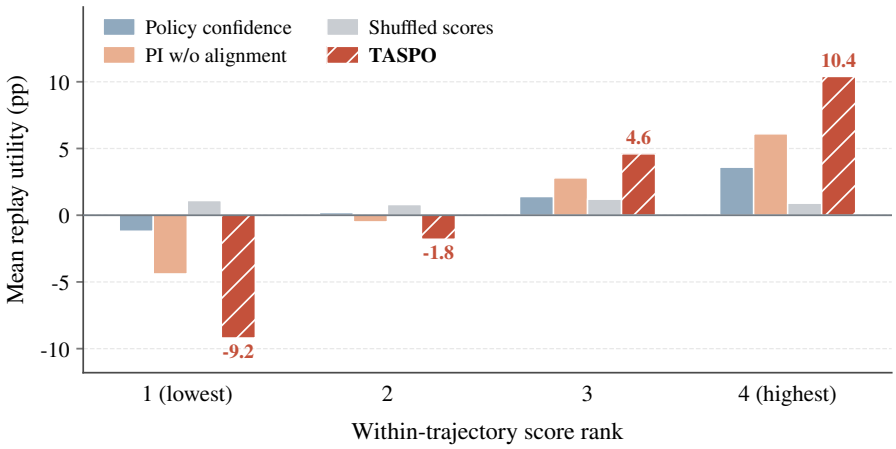


Figure D.2: **Replay-estimated relative utility by within-trajectory score rank.** Each trajectory's four selected decision points are ordered by each scoring signal. Bars report mean relative replay utility at each rank, in percentage points, from the lowest rank (1) to the highest (4). These ranks are not pooled population quantiles. Error bars denote 95% bootstrap confidence intervals. The raw-sibling control is labeled "PI w/o alignment" in the plot.

Results and interpretation. TASPO has an observed Spearman correlation of 0.41 and pairwise accuracy of 69.6%, compared with 0.08 and 52.8% for policy confidence and 0.24 and 60.7% for raw successful-sibling PI. Its top-to-bottom replay-utility gap is 19.6 percentage points. These results show greater ranking agreement with the replay estimates than the listed controls and support aligned likelihood shifts as a relative credit signal in this diagnostic.

The diagnostic does not establish that d_{it} is an unbiased advantage estimator or an estimate of absolute action utility. Replay outcomes are sampled, the alternative set covers only part of the action space, and pairwise accuracy is evaluated on a selected subset of action pairs. The additional replay interactions serve only this diagnostic and are not required by TASPO's training-time scoring and credit-redistribution procedure.

D.5 Computational Cost

Additional computation on existing rollouts. TASPO adds guidance extraction and alignment, followed by PI-conditioned scoring of already collected responses. Original-view log probabilities are reused when they come from the same frozen rollout snapshot and use the same scoring definition. Neither PI construction nor rescoring itself requires additional environment rollouts; the replay study in Section D.4 is a separate diagnostic.

With N_{upd} updates, B_{task} tasks per update, and K rollouts per task, the scheduled trajectory count is

$$N_{\text{upd}} B_{\text{task}} K.$$

The reported settings yield 19,200 scheduled trajectories for ALFWorld and WebShop and 153,600 for Search-QA.

We profile an ALFWorld run with Qwen2.5-3B-Instruct on eight H800 GPUs through update 150. Including one replayed batch, the logs contain 151 completed batches, 19,328 trajectories, 508,704 environment steps, and 2,974 analyzer request attempts.

Figure D.3 reports selected stage runtimes. PI construction takes 4.50 hours, and PI-conditioned scoring takes 0.23 hours, approximately 5.4 seconds per completed batch. Guidance construction therefore accounts for more recorded time than the additional scoring pass. The profiles use different hardware configurations and omit several stages, so they locate the added computation rather than establish a hardware-matched end-to-end speedup or total-cost advantage.

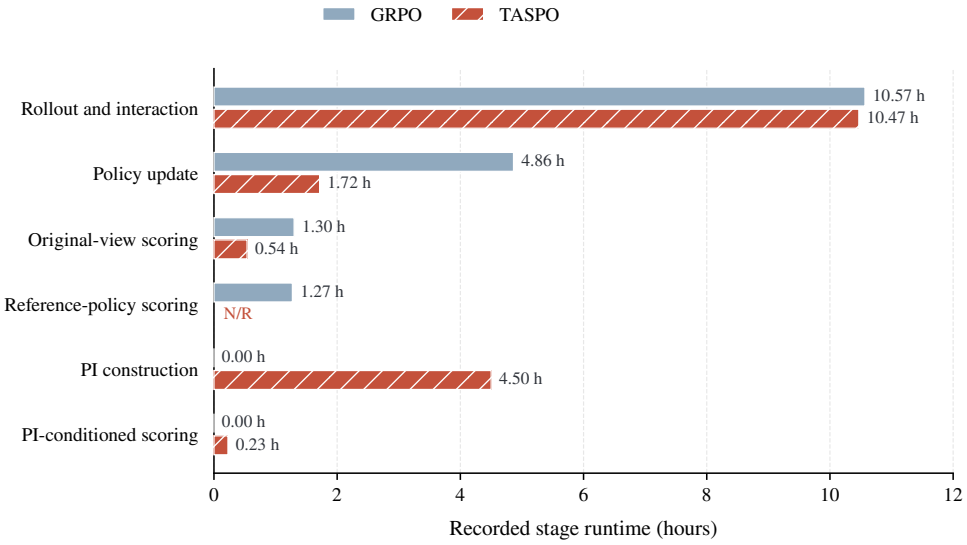


Figure D.3: **Recorded stage runtimes for two ALFWorld training runs.** Both use Qwen2.5-3B-Instruct through update 150. GRPO uses four GPUs of an unconfirmed model; TASPO uses eight H800 GPUs. Bars report selected stages rather than total wall-clock runtime. N/R denotes a stage that was not separately reported. Initialization, evaluation, and checkpoint saving are excluded. The hardware difference precludes a controlled speed comparison.

D.6 Comparison with Additional Credit-Assignment Methods

Tables D.3 and D.4 provide additional comparisons with CRAFT (Meng & Chen, 2026) and AT^{2}PO (Zong et al., 2026) under the respective reported experimental protocols. These comparisons complement the controlled ablations above; differences in experimental protocols should be considered when interpreting absolute performance.

TASPO has higher reported ALFWorld success than CRAFT at all three displayed backbone sizes. On WebShop, the comparison is mixed: TASPO has the higher score with Qwen2.5-3B, whereas CRAFT has higher scores with Qwen2.5-7B and Qwen3-1.7B. On Search-QA, TASPO's macro-average is 49.8%, compared with 48.5% for AT^{2}PO. The dataset-level pattern is uneven: AT^{2}PO is higher on five of the seven datasets, whereas TASPO is higher on TriviaQA and Bamboogle. These results provide cross-protocol context with task- and backbone-dependent differences. Evidence for the specific effects of target alignment and supervision integration comes from the controlled comparisons in Sections D.1 and D.2.

Table D.3: **Additional ALFWorld and WebShop results across three backbones.** ALFWorld reports macro-averaged task-family success; WebShop reports normalized task score and success rate, all in percent. TASPO values reproduce the main results table. CRAFT success rates on WebShop are not reported in this comparison (—). Bold metric values indicate the higher result where both methods report the metric. Comparisons follow the respective reported protocols.

Method	**ALFWorld**	**WebShop**	
	ALF Avg.	Score	Succ.
Qwen2.5-3B			
CRAFT	85.0	87.3	—
TASPO	**86.2**	**88.5**	78.1
Qwen2.5-7B			
CRAFT	87.8	**91.0**	—
TASPO	**90.1**	87.7	76.4
Qwen3-1.7B			
CRAFT	57.3	**82.3**	—
TASPO	**69.6**	77.6	67.1

Table D.4: **Additional Search-QA results with Qwen2.5-7B.** Values are exact-match accuracy in percent under each work's reported experimental protocol. Avg. is the unweighted mean of the seven displayed dataset scores. The comparison provides cross-protocol context rather than controlled attribution of performance differences.

Method	NQ	Triv	Pop	Hotp	2Wk	MuS	Bam	**Avg.**
AT^{2}PO	**52.9**	64.9	**50.4**	**49.6**	**48.0**	**22.6**	51.2	48.5
TASPO	49.6	**65.9**	49.3	45.1	43.3	18.4	**76.7**	**49.8**

D.7 Training Dynamics

Figure D.4 compares redistribution budgets $\epsilon_w \in \{0.4, 0.6, 0.8\}$ on ALFWorld with Qwen2.5-3B-Instruct over 150 policy updates. Across these settings, training reward and validation performance generally improve, and response lengths level off after an initial increase. Policy entropy remains nonzero, while entropy and gradient norms decline with local fluctuations.

The default budget $\epsilon_w = 0.4$ reaches the highest final macro-averaged success among the displayed settings: 86.2%, compared with 84.2% and 83.5% for $\epsilon_w = 0.6$ and 0.8, respectively. This comparison supports retaining $\epsilon_w = 0.4$ as the default within the displayed range.

Panel (e) reports the global token-weighted advantage mean. This logging statistic differs from the per-trajectory, turn-averaged scalar credit preserved by TASPO. Its deviation from zero therefore does not contradict the conservation result. The curves characterize observed training dynamics rather than independently establish a reduction in gradient variance or a convergence guarantee.

E Qualitative Analysis

The cases below illustrate two decision patterns relevant to credit assignment: restoring an execution precondition after an ineffective action, and revising a permitted purchase that cannot satisfy the task. The displayed interactions do not contain per-turn training scores or credit weights. We therefore distinguish the visible behavior from its interpretation under the credit-allocation scheme.

Case 1: Restoring an execution precondition on ALFWorld. The task requires heating a cup and placing it in a cabinet. At the decision point in Figure E.1, the cup has been heated, but an attempt to open `cabinet 1` returns "Nothing happens." Repeating the same command under an unchanged state leaves the execution condition unresolved. The relevant distinction is whether the agent retries immediately or first changes its interaction state.

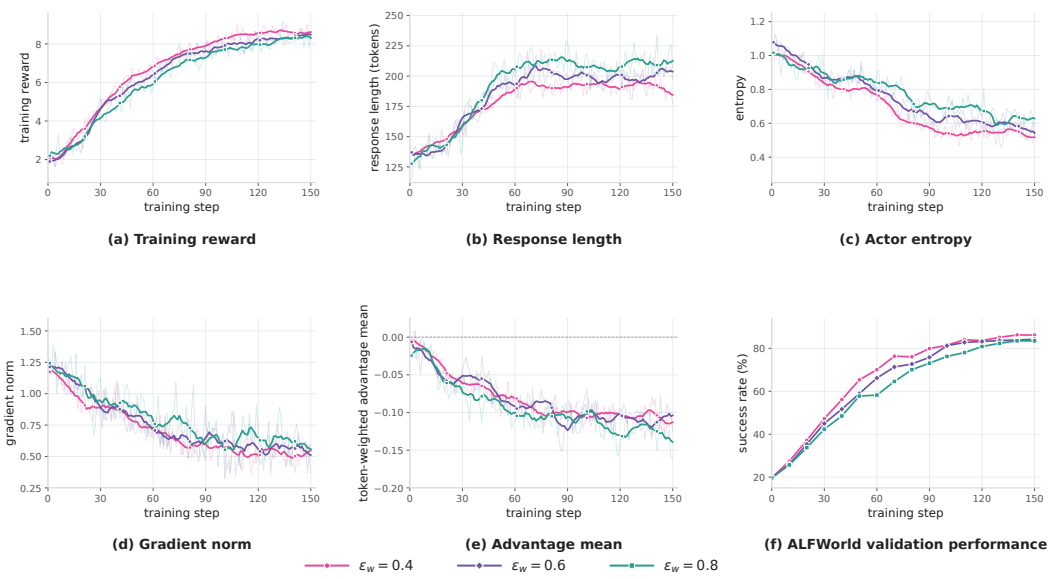


Figure D.4: **Training dynamics across redistribution budgets on ALFWorld.** Qwen2.5-3B-Instruct is trained for 150 policy updates with $\epsilon_w \in \{0.4, 0.6, 0.8\}$. Panels show (a) training reward, (b) response length, (c) policy entropy, (d) gradient norm, (e) the global token-weighted advantage mean, and (f) macro-averaged validation success. Panel (e) uses a different aggregation from the per-trajectory credit-preservation constraint.

The recorded TASPO recovery branch executes `go to cabinet 1` before retrying `open cabinet 1`, after which the opening succeeds. Subsequent interactions lead to placement in `cabinet 16` and task completion. The left branch contrasts this recovery with repeated opening attempts; the displayed repetitions alone do not establish budget exhaustion.

Credit-assignment interpretation. If the successful trajectory receives a positive group-relative advantage, uniform outcome credit assigns the same scalar advantage to its ineffective opening attempt, the navigation preceding the successful retry, and the eventual placement. TASPO is designed to distinguish their relative support while preserving the advantage's sign and trajectory mean. Reducing the weight of a less-supported action on a positive-advantage trajectory reduces its positive credit; it does not turn that credit into a penalty. The recorded recovery illustrates the behavior this distinction is intended to support, without identifying which training-time weight produced it.

Case 2: Revising a constraint-violating choice on WebShop. The task requests a blue 32 oz water bottle priced below $30.00. In Figure E.2, the first product satisfies the color and price requirements but offers only a 16 oz capacity. Its detail page therefore provides enough information to reject it. Completing a permitted purchase at this point would still leave a task requirement unmet.

In the displayed comparison, GRPO selects the requested color and purchases the 16 oz candidate. TASPO returns to search and reformulates its query around the required capacity, then selects a blue 32 oz bottle priced at $27.99. Returning to search does not immediately complete the task, but it enables the subsequent selection of a suitable product. The visible difference is the revision of an intermediate choice after observing an incompatible attribute.

Credit-assignment interpretation. A terminal outcome does not distinguish the roles of product inspection, a constraint-violating purchase, and a return to search. Target alignment is intended to make guidance conditional on the requested attributes and the options available in the recorded interaction. The case illustrates why successful execution of an interface action need not imply progress toward the task goal.

Task and shared decision point

Task: Heat some cup and put it in cabinet.

The recorded prefix has reached `microwave 1` and heated `cup 3`.

Environment (Step 12)

You heat the cup 3 using the microwave 1.

Agent (Step 12)

`open cabinet 1`

Environment (Step 13)

Nothing happens.

Repeated-action continuation	Recorded recovery continuation
Agent (Step 13) `open cabinet 1`	**Agent (Step 13)** `go to cabinet 1`
Environment (Step 14) Nothing happens.	**Environment (Step 14)** You arrive at cabinet 1. The cabinet 1 is closed.
Agent (Step 14) `open cabinet 1`	**Agent (Step 14)** `open cabinet 1`
Environment (Step 15) Nothing happens.	**Environment (Step 15)** You open the cabinet 1. The cabinet 1 is open. In it, you see nothing.
Unresolved execution condition The agent repeats the operation without first navigating to the cabinet. Under the assumed unchanged state, these attempts make no progress. Continued repetition would consume the interaction budget while leaving the placement goal unmet.	**Steps 15–18: recorded actions, condensed** The agent visits `countertop 1`, checks its inventory, navigates to `cabinet 16`, and opens it. **Environment (Step 19)** You open the cabinet 16. The cabinet 16 is open. In it, you see nothing. **Agent (Step 19)** `move cup 3 to cabinet 16` **Recorded outcome** Task success: **True**.

Figure E.1: **Repeated opening attempts and recorded recovery on ALFWorld.** The left branch shows repeated opening attempts under an unchanged execution condition. The recorded TASPO recovery on the right navigates to `cabinet 1` before retrying, and subsequently completes the task by placing the heated cup in `cabinet 16`. The right branch contains selected interactions from a Qwen2.5-3B-Instruct rollout; Steps 15–18 are condensed.

AI USE STATEMENT

We used AI tools to assist with language polishing to enhance the clarity and readability of the manuscript. We conceived the core research concepts, designed the methodology, and established the main scientific arguments. We reviewed all AI-assisted revisions and bear full responsibility for the final manuscript.

Environment (Step 1)

Task: Find a blue 32 oz water bottle for less than $30.00.

Current observation: Search.

Agent (Step 1)

```
search[blue 32 oz water bottle under 30 dollars]
```

Environment (Step 2)

Search results include:

Product A: Water Bottle. Price: $24.99.

Agent (Step 2)

```
click[Product A]
```

Environment (Step 3)

Product A

Price: $24.99. Available colors: blue, black. Available capacity: **16 oz only**.

[Back to Search] [Buy Now]

GRPO

Agent (Step 3)

```
click[blue]
```

Environment (Step 4)

Product A.
Selected color: blue.
Capacity: **16 oz**.
Price: $24.99.

[Back to Search] [Buy Now]

Agent (Step 4)

```
click[Buy Now]
```

Purchase outcome

Purchased: blue, 16 oz, $24.99.

The required 32 oz capacity is not met.

TASPO

Agent (Step 3)

```
click[Back to Search]
```

Environment (Step 4)

Search.

Agent (Step 4)

```
search[32 ounce capacity blue water bottle]
```

Environment (Step 5)

Search results include:

Product B: 32 oz Water Bottle. Price: $27.99.

Agent (Step 5)

```
click[Product B]
```

Environment (Step 6)

Product B
Price: $27.99.
Available colors: blue, black.
Capacity: **32 oz**.

[Back to Search] [Buy Now]

Agent (Step 6)

```
click[blue]
```

Environment (Step 7)

Product B.
Selected color: blue.
Capacity: **32 oz**.
Price: $27.99.

[Back to Search] [Buy Now]

Agent (Step 7)

```
click[Buy Now]
```

Purchase outcome

Purchased: blue, 32 oz, $27.99.

All stated task constraints are met.

Figure E.2: **Revising product selection after a capacity mismatch on WebShop.** The first candidate meets the color and price requirements but offers 16 oz rather than the requested 32 oz. The displayed GRPO continuation purchases this candidate; TASPO returns to search, revises its query, and selects a product satisfying the stated constraints.