SPARSE WHERE, DENSE WITHIN: CONTRASTIVE PRIVILEGED ACTION DISTILLATION FOR LONG-HORIZON AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Training long-horizon language model agents from terminal rewards is challenging because task outcomes provide limited guidance for individual decisions. On-policy distillation can supply dense token-level supervision using privileged training information. However, dense feedback alone does not identify which decisions warrant additional guidance, and positive guidance can conflate action preferences with the effects of added context. We introduce Contrastive Privileged Action Distillation (CPAD), which uses action comparisons supported by rollout outcomes to determine both where and how to distill. CPAD matches repeated decision states across on-policy trajectories and selects only those where downstream outcomes support a preferred action over an inferior alternative. At each selected decision, the same model re-scores its sampled response under two contexts that differ only in the guiding action. Their token-level log-probability difference augments the outcome-based advantage, making supervision selective across decisions and dense within each selected response. Privileged guidance is used only during training. Experiments on ALFWorld and WebShop with Qwen2.5 models at 1.5B, 3B, and 7B scales show competitive final performance against structured credit-assignment methods. Compared with a matched outcome-based RL baseline, CPAD improves final success and the area under the 150-update validation success curve in all six settings on ALFWorld seen tasks and WebShop. At 150 updates, final success increases by up to 24.4 and 11.4 percentage points on the two benchmarks, respectively, with 24–49% fewer training interactions.

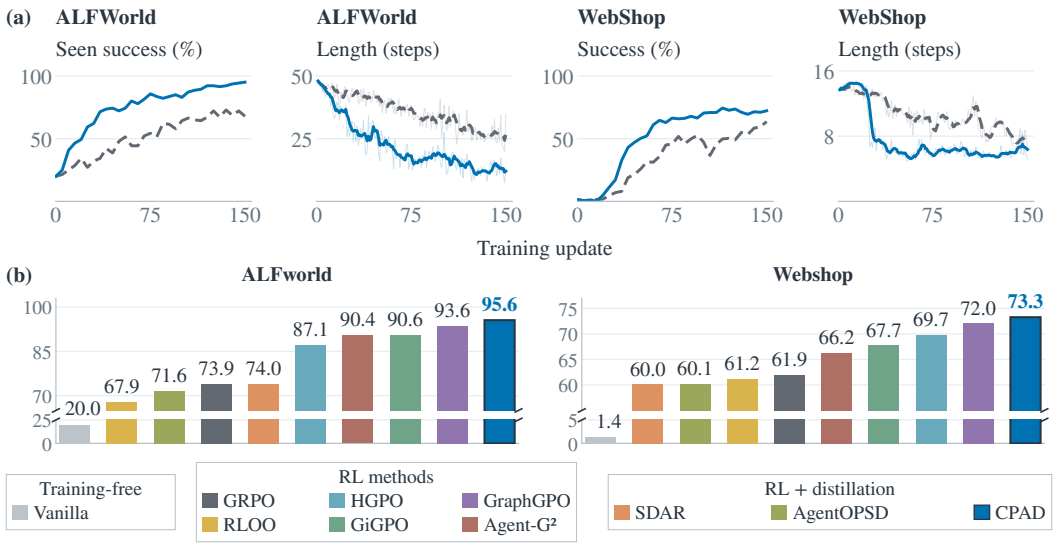

Figure 1: All experiments shown use Qwen2.5-3B-Instruct. (a) Validation success rate and mean training trajectory length for GRPO and CPAD; (b) Final success rates across methods.

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.

1 INTRODUCTION

Language model agents solve tasks through sequences of actions, from manipulating household objects to purchasing products online (Shridhar et al., 2021; Yao et al., 2023). Success requires coordinating decisions across many turns. Learning this long-horizon capability therefore requires identifying which earlier decisions contributed to the final outcome.

Outcome-based reinforcement learning (RL), including RLOO and GRPO, provides a task-level objective (Ahmadian et al., 2024; Shao et al., 2024), but delayed rewards offer limited guidance to individual decisions and response tokens. On-policy distillation (OPD) complements RL with token-level teacher feedback on the policy's own sampled responses. In on-policy self-distillation (OPSD), the same model serves as teacher with additional training-only information; re-scoring a stored response reveals how that information changes support for each token. Recent methods draw such supervision from retrieved skills, trajectory hindsight, evolving analyzers, or successful peer trajectories (Lu et al., 2026; Yang et al., 2026b; Wu et al., 2026a; Wang et al., 2026a; Zhang et al., 2026).

Two questions arise for long-horizon agents. **Where should distillation act?** A trajectory contains routine operations, recoverable detours, and decisions at which different actions are associated with different downstream outcomes. Distillation throughout the trajectory can reshape decisions for which no local comparison supports a change: token-level density alone does not establish where guidance is warranted. Existing work studies the reliability of teacher–student gaps, policy-synchronized knowledge, and step- or turn-level aggregation (Lu et al., 2026; Yang et al., 2026b; Wu et al., 2026a; Wang et al., 2026a; Zhang et al., 2026). We ask whether the current rollout batch supports an actionable comparison at the same decision context.

How should selected decisions be guided? Positive privileged guidance alone can yield unreliable supervision (Pan et al., 2026; Wu et al., 2026b). A preferred action suggests useful behavior, but its teacher score also reflects the added context. Contrasting preferred and inferior action guidance provides a relative signal for the same sampled response.

To address these two questions jointly, we introduce **Contrastive Privileged Action Distillation (CPAD)**, which uses action comparisons supported by rollout outcomes to select decisions for distillation and construct paired action guidance. **First, select supported decision comparisons.** On-policy trajectories for the same task can reach equivalent decision contexts, take different actions, and receive different downstream outcomes. CPAD matches these contexts and acts only when their outcomes support a preferred action and an inferior alternative. Groups without a supported preference retain the outcome-based RL objective, making CPAD selective across decisions.

Second, combine positive and negative action information. For each selected decision, CPAD inserts the preferred and inferior actions into the same prompt template to construct two privileged contexts. The frozen pre-update actor re-scores the same sampled response under both. Their token-level log-probability difference becomes a bounded local correction to the outcome advantage: supervision is sparse across decisions and dense within each selected response. Privileged actions are used only during training; rollout generation and deployment use the ordinary interaction context.

We evaluate CPAD on ALFWorld and WebShop with Qwen2.5-Instruct models at 1.5B, 3B, and 7B scales. CPAD improves the area under the validation curve and final success over matched GRPO in all six settings, with 24–49% fewer training interactions at 150 updates. Figure 1 illustrates the 3B results; Section 4.4 reports interaction savings and training time, including privileged re-scoring. Our contributions are:

- **A selective view of agentic OPD.** We use preference-bearing matched decision groups to determine where dense privileged supervision should modify the outcome advantage.
- **Matched-decision contrastive teachers.** CPAD uses the same local evidence to construct preferred- and inferior-action guidance, yielding dense token-level corrections on selected on-policy responses.
- **Cross-scale evaluation and mechanism diagnostics.** Experiments across two benchmarks and three model scales show consistent learning-curve gains over GRPO; branch ablations, trajectory lengths, and reference KL diagnostics characterize the paired signal and training behavior.

2 PRELIMINARIES

An agent solves task x through multiple interaction rounds. At step t of rollout i, the context $h^0_{i,t}$ contains the task, observation $o_{i,t}$, and available interaction history. The policy samples response $y_{i,t}$, from which action $a_{i,t}$ is parsed and executed. A trajectory $\tau_i = ((o_{i,t}, y_{i,t}, a_{i,t}))_{t=1}^{T_i}$ receives terminal reward R_i. Let $\mathcal{K}_{i,t}$ index valid response tokens $y_{i,t,k}$, with prefix $y_{i,t,<k}$, and let $\bar{\theta}$ denote the pre-update policy parameters.

2.1 GROUP RELATIVE POLICY OPTIMIZATION (GRPO)

GRPO (Shao et al., 2024) estimates advantages from a group of N trajectories sampled for the same task under $\pi_{\bar{\theta}}$. It normalizes each terminal reward using the group's reward mean and standard deviation:

$$A_i^{\mathrm{GRPO}} = \frac{R_i - \mathrm{mean}_{j=1}^{N} R_j}{\mathrm{std}_{j=1}^{N} R_j + \epsilon}, \tag{1}$$

where $\epsilon > 0$ stabilizes normalization. This trajectory-level advantage is broadcast to every response token. For the trainable policy π_θ, the token-level importance ratio is

$$\rho_{i,t,k}(\theta) = \frac{\pi_\theta(y_{i,t,k} \mid h^0_{i,t}, y_{i,t,<k})}{\pi_{\bar{\theta}}(y_{i,t,k} \mid h^0_{i,t}, y_{i,t,<k})}. \tag{2}$$

The clipped objective, written as a loss to minimize, is

$$\begin{aligned}\mathcal{L}_{\mathrm{GRPO}}(\theta) = & -\mathbb{E}_{i,t,k}\big[\min\big(\rho_{i,t,k}(\theta)A_i^{\mathrm{GRPO}},\ \mathrm{clip}(\rho_{i,t,k}(\theta), 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}})\, A_i^{\mathrm{GRPO}}\big)\big] \\ & + \beta\, \mathbb{E}_{i,t,k} D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}).\end{aligned} \tag{3}$$

Here the expectation averages over tasks and sampled trajectories, with valid tokens averaged within each trajectory. Each KL term compares the next-token distributions at the same context and response prefix; π_{ref} is a fixed reference policy, β is its regularization weight, and ϵ_{clip} is the ratio-clipping radius. The shared outcome advantage provides trajectory-level supervision without resolving individual token contributions.

2.2 ON-POLICY DISTILLATION (OPD)

OPD trains a student on its own generated responses using teacher supervision. The student samples under $h^0_{i,t}$; a teacher π_T evaluates the same response prefixes under context $h^T_{i,t}$, which may include additional information. A common distillation objective minimizes the reverse KL from student to teacher (Lu et al., 2026):

$$\mathcal{L}_{\mathrm{OPD}}(\theta) = \mathbb{E}_{i,t,k}\big[D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid h^0_{i,t}, y_{i,t,<k}) \,\big\|\, \pi_T(\cdot \mid h^T_{i,t}, y_{i,t,<k})\big)\big]. \tag{4}$$

The expectation is over student-generated contexts and prefixes, with the teacher held fixed during each update. At the pre-update policy, a token sampled from $\pi_{\bar{\theta}}$ gives the teacher–student log-probability gap

$$d^{\mathrm{OPD}}_{i,t,k} = \log \pi_T(y_{i,t,k} \mid h^T_{i,t}, y_{i,t,<k}) - \log \pi_{\bar{\theta}}(y_{i,t,k} \mid h^0_{i,t}, y_{i,t,<k}). \tag{5}$$

Its negative is a single-sample estimate of the conditional reverse KL at $\theta = \bar{\theta}$, and the detached gap can serve as token-level guidance. In on-policy self-distillation (OPSD), the teacher can be a frozen copy of the student conditioned on additional skills or hindsight (Lu et al., 2026; Yang et al., 2026b; Wu et al., 2026a).

3 CONTRASTIVE PRIVILEGED ACTION DISTILLATION

CPAD uses action comparisons within the current rollout batch to decide *where* distillation should act and *how* it should guide the policy. Starting from on-policy trajectories and binary success outcomes $R_i \in \{0, 1\}$, it selects matched decisions with supported action preferences (Figure 2(a–b)). At these decisions, the frozen actor re-scores the stored response under preferred- and inferior-action guidance; their token-level score difference corrects the outcome advantage used for policy optimization (c–e). Algorithm 1 in Appendix A summarizes the training procedure.

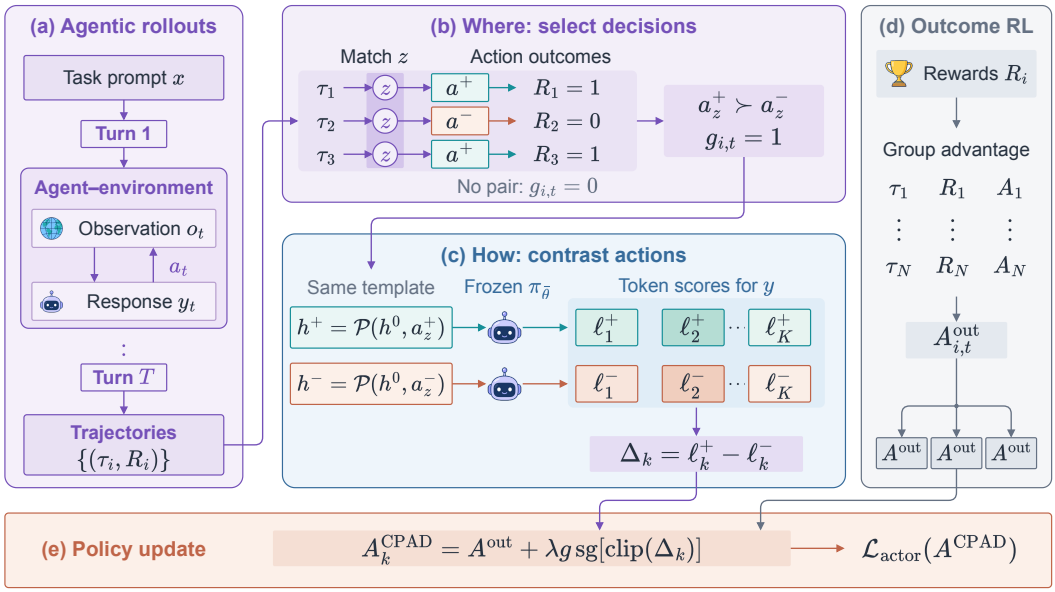

Figure 2: **Overview of CPAD. (a)** On-policy rollouts provide trajectories and outcomes. **(b)** Matched decision groups identify supported action preferences. **(c)** The same frozen actor re-scores each selected response under preferred- and inferior-action guidance. **(d–e)** The token-level score contrast adds a local correction to the outcome advantage for policy optimization.

3.1 Preference-Bearing Decision Groups

For each task x, CPAD starts from on-policy trajectories $\tau_i = ((o_{i,t}, y_{i,t}, a_{i,t}))_{t=1}^{T_i}$ with terminal rewards R_i. Each (i,t) denotes a decision occurrence. To compare actions across trajectories, CPAD groups these occurrences by their normalized structured decision states $\xi_{i,t}$, using the matching key $z_{i,t} = \text{key}(x, \xi_{i,t})$. This matching representation is distinct from the full model context $h_{i,t}^0$. Occurrences sharing a key form $\mathcal{G}_z = \{(i,t) : z_{i,t} = z\}$.

Within each matched group, we estimate how often each executed action occurs on a successful trajectory. Partitioning $\mathcal{G}_z$ by action gives

$$\mathcal{G}_z(a) = \{(i,t) \in \mathcal{G}_z : a_{i,t} = a\}, \qquad \widehat{r}_z(a) = \frac{\sum_{i \in \mathcal{I}_z(a)} R_i}{|\mathcal{I}_z(a)|}, \quad \mathcal{I}_z(a) = \{i : \exists t,\ (i,t) \in \mathcal{G}_z(a)\}. \tag{6}$$

The trajectory-level set $\mathcal{I}_z(a)$ prevents repeated visits to the same state from overcounting an action. The default selector requires at least two trajectories per group and at least one supporting trajectory per retained action. Each group must contain at least two executed actions, including one observed on a successful trajectory.

The selector first identifies a preferred action a_z^+ using the empirical success rate $\widehat{r}_z(a)$ as its primary criterion. When several actions share the highest rate, it uses successful continuation quality—fewer remaining decisions and lower future penalties—as secondary evidence. The preferred action must have successful support and be the unique supported, non-dominated top action; otherwise CPAD abstains.

To form a contrastive pair, the selector then chooses an inferior action a_z^- from sufficiently supported alternatives with either a lower success rate or worse successful continuation quality. It ranks these alternatives by lower success first, then by longer and more heavily penalized continuations, with a deterministic string tie-break. If no such alternative exists, CPAD again abstains.

Every valid group is retained, and its pair (a_z^+, a_z^-) is attached to all group occurrences. The decision-level distillation gate $g_{i,t} \in \{0,1\}$ enables the paired token-level correction at these decisions: $g_{i,t} = 1$ for retained group occurrences and $g_{i,t} = 0$ otherwise. The attached action pairs supply the guidance for the two privileged contexts constructed in Section 3.2.

3.2 Paired Privileged Action Distillation

For a selected occurrence, CPAD constructs preferred-action context $h^+_{i,t}$ and inferior-action context $h^-_{i,t}$ using the same prompt template:

$$h^+_{i,t} = \mathcal{P}(h^0_{i,t}, a^+_z), \qquad h^-_{i,t} = \mathcal{P}(h^0_{i,t}, a^-_z).$$

Both branches present the inserted action as guidance for the current decision using identical wording. Their polarity remains in trainer metadata, so the prompt text differs only in the inserted action. Appendix G provides the full interaction and privileged prompt templates.

The sampled response and its token prefixes are held fixed while the pre-update actor $\pi_{\bar{\theta}}$ evaluates each context by teacher forcing. For an arbitrary scoring context h, define

$$\ell_{i,t,k}(h) = \log \pi_{\bar{\theta}} \left(y_{i,t,k} \mid h, y_{i,t,<k} \right). \tag{7}$$

The ordinary and privileged views give

$$\ell^b_{i,t,k} = \ell_{i,t,k}(h^b_{i,t}), \qquad b \in \{0, +, -\}. \tag{8}$$

The scores in Equation 8 form two one-view components: the preferred-view component $d^+_{i,t,k} = \ell^+_{i,t,k} - \ell^0_{i,t,k}$ and the inferior-control component $d^-_{i,t,k} = \ell^0_{i,t,k} - \ell^-_{i,t,k}$. Their equal-weight sum defines one raw contrast:

$$\Delta_{i,t,k} = d^+_{i,t,k} + d^-_{i,t,k} = \ell^+_{i,t,k} - \ell^-_{i,t,k}. \tag{9}$$

The ordinary-context score cancels, so CPAD computes the contrast with two teacher-forced passes of the same frozen actor on the stored response.

CPAD gates, clips, and detaches the raw contrast before optimization:

$$\widetilde{\Delta}_{i,t,k} = g_{i,t} \ \mathrm{sg}\left[\mathrm{clip}\left(\Delta_{i,t,k}, -c_\Delta, c_\Delta \right) \right]. \tag{10}$$

Here c_Δ is the contrast bound and sg denotes stop-gradient. In Equation 10, selection is decision-level, while the correction is defined for every $k \in \mathcal{K}_{i,t}$. The CPAD+ and CPAD− ablations retain only $d^+_{i,t,k}$ and $d^-_{i,t,k}$, respectively; canonical CPAD uses their equal-weight sum. Appendix I provides a theoretical analysis.

3.3 Outcome-Anchored Policy Optimization

CPAD and the matched GRPO baseline share a decision-level outcome advantage:

$$A^{\mathrm{out}}_{i,t} = \frac{\widetilde{R}_{i,t} - \mu_x}{\sigma_x + \epsilon}, \tag{11}$$

where $\widetilde{R}_{i,t}$ is the scaled terminal reward with a local invalid-action penalty. The mean μ_x and sample standard deviation σ_x are computed over all decision occurrences in the task's rollout group. Appendix A gives the reward and normalization details. This advantage is broadcast to valid response tokens, and the contrast adds a local correction:

$$A^{\mathrm{CPAD}}_{i,t,k} = A^{\mathrm{out}}_{i,t} + \lambda \widetilde{\Delta}_{i,t,k}, \qquad k \in \mathcal{K}_{i,t}, \tag{12}$$

where λ controls correction strength. If $g_{i,t} = 0$, then Equation 12 reduces to $A^{\mathrm{CPAD}}_{i,t,k} = A^{\mathrm{out}}_{i,t}$, so CPAD leaves the outcome anchor unchanged. At selected decisions, the additive correction can change the sign of the outcome advantage.

Let $\mathcal{L}_{\mathrm{actor}}(A)$ denote the shared actor loss for token advantages A. It uses the clipped surrogate in Equation 3 with dual clipping, reference KL regularization, and an entropy bonus. CPAD optimizes $\mathcal{L}_{\mathrm{actor}}(A^{\mathrm{CPAD}})$, while outcome-only training uses $\mathcal{L}_{\mathrm{actor}}(A^{\mathrm{out}})$. Appendix H reports the optimization hyperparameters.

Privileged actions are used only during training. Rollout generation, validation, and deployment use the original interaction context.

Table 1: Full-set validation results. VANILLA is GRPO at update 0; all other cells use update 150. All values are percentages. ALFWorld reports family and aggregate success on 140 `val_seen` and 134 `val_unseen` tasks; WebShop reports the official partial-credit **Score** and exact **Succ.** on 500 fixed goals. Green and red cells mark the highest and second-highest available values within each model scale, respectively.

	ALFWorld Seen							ALFWorld Unseen							WebShop	
Method	Pick	Look	Clean	Heat	Cool	Pick2	Succ.	Pick	Look	Clean	Heat	Cool	Pick2	Succ.	Score	Succ.
Qwen2.5-1.5B-Instruct																
Vanilla	11.4	15.4	7.4	0.0	8.0	4.2	7.9	20.8	0.0	6.5	13.0	0.0	0.0	7.5	20.0	4.8
GRPO	73.1	53.8	67.4	67.5	69.6	50.8	65.1 ± 1.8	68.3	70.0	61.3	70.4	55.2	32.9	60.7 ± 2.1	74.5 ± 1.0	61.0 ± 1.4
RLOO	88.0	75.4	70.4	67.5	68.0	52.5	71.4 ± 1.7	70.8	75.6	63.2	81.7	67.6	41.2	67.3 ± 2.3	82.3 ± 0.6	66.6 ± 1.1
HGPO	92.0	70.8	94.1	91.2	84.8	71.7	85.6 ± 1.3	**85.8**	96.7	84.5	79.1	76.2	71.8	82.5 ± 1.9	82.5 ± 0.8	67.3 ± 0.6
GiGPO	92.6	69.2	93.3	92.5	**87.2**	78.3	87.1 ± 0.7	76.7	91.1	83.2	88.7	85.7	65.9	82.2 ± 0.3	83.9 ± 0.4	69.7 ± 0.3
GraphGPO	94.3	72.3	94.8	**93.8**	86.4	**86.7**	**89.6 ± 1.2**	72.5	**98.9**	**91.0**	85.2	**95.2**	**88.2**	**88.1 ± 1.7**	85.2 ± 0.7	**74.2 ± 0.8**
SDAR	84.0	67.7	78.5	70.0	72.8	67.5	75.0 ± 3.1	70.0	87.8	77.4	76.5	75.2	54.1	74.0 ± 2.2	78.1 ± 0.8	61.4 ± 0.8
AgentOPSD	74.9	**76.9**	71.1	71.2	70.4	58.3	70.3 ± 3.2	69.2	82.2	67.1	77.4	75.2	58.8	71.5 ± 2.8	75.6 ± 0.8	59.5 ± 0.9
Agent-G^2	93.7	55.4	**97.8**	**93.8**	**87.2**	77.5	87.0 ± 0.9	84.2	41.1	87.7	80.9	81.9	56.5	74.8 ± 1.9	**85.4 ± 0.6**	73.0 ± 0.8
CPAD	**94.9**	75.4	97.0	**93.8**	**87.2**	80.8	**89.6 ± 0.6**	73.3	70.0	85.2	**91.3**	84.8	85.9	82.1 ± 1.2	85.1 ± 0.5	69.5 ± 0.6
CPAD+	42.9	46.2	31.1	2.5	20.8	23.3	29.0 ± 1.8	23.3	22.2	26.5	2.6	11.4	8.2	16.6 ± 1.5	69.4 ± 2.1	49.2 ± 2.1
CPAD−	14.9	3.1	1.5	17.5	2.4	3.3	7.3 ± 2.8	20.0	0.0	0.0	3.5	1.9	0.0	4.5 ± 1.4	0.2 ± 0.2	0.1 ± 0.1
Qwen2.5-3B-Instruct																
Vanilla	48.6	30.8	7.4	0.0	16.0	4.2	20.0	50.0	11.1	6.5	21.7	9.5	5.9	17.9	5.6	1.4
GRPO	98.3	63.1	75.6	60.0	64.8	60.8	73.9 ± 1.7	85.8	64.4	75.5	59.1	69.5	51.8	69.1 ± 2.6	80.5 ± 1.2	61.9 ± 1.0
RLOO	93.1	63.1	75.6	38.8	61.6	50.8	67.9 ± 2.1	78.3	60.0	69.0	44.3	66.7	43.5	61.6 ± 2.8	75.0 ± 0.9	61.2 ± 1.6
HGPO	98.9	83.1	95.6	86.2	76.0	75.0	87.1 ± 1.0	**88.3**	86.7	88.4	80.9	95.2	84.7	87.5 ± 1.5	84.6 ± 0.6	69.7 ± 0.7
GiGPO	**100.0**	87.7	96.3	72.5	87.2	87.5	90.6 ± 2.5	81.7	74.4	89.0	83.5	**98.1**	90.6	86.4 ± 1.9	82.3 ± 0.6	67.7 ± 0.7
GraphGPO	**100.0**	**92.3**	**97.0**	**93.8**	88.0	86.7	93.6 ± 0.9	86.7	97.8	89.0	80.0	95.2	**94.1**	89.9 ± 1.2	84.7 ± 0.9	72.0 ± 1.2
SDAR	93.7	53.8	89.6	60.0	62.4	60.0	74.0 ± 2.7	76.7	55.6	80.0	75.7	75.2	56.5	71.6 ± 2.6	76.8 ± 1.2	60.0 ± 1.3
AgentOPSD	96.6	61.5	74.8	46.2	64.8	60.8	71.6 ± 2.2	81.7	60.0	69.7	51.3	72.4	54.1	65.8 ± 3.0	77.0 ± 1.6	60.1 ± 1.1
Agent-G^2	**100.0**	81.5	96.3	88.8	91.2	75.0	90.4 ± 2.6	81.7	67.8	**92.9**	**87.8**	96.2	75.3	84.9 ± 1.0	75.6 ± 0.8	66.2 ± 1.0
CPAD	**100.0**	87.7	96.3	91.2	**96.0**	**95.0**	**95.6 ± 0.8**	85.8	**98.9**	90.3	84.3	95.2	88.2	**90.1 ± 0.6**	**87.3 ± 0.6**	73.3 ± 1.0
CPAD+	72.0	69.2	71.9	83.8	73.6	60.0	71.3 ± 1.3	58.3	88.9	61.3	61.7	82.9	61.2	67.9 ± 3.0	83.7 ± 0.4	**73.4 ± 0.6**
CPAD−	8.6	21.5	0.7	1.2	0.0	4.2	5.1 ± 1.8	1.7	21.1	0.0	0.0	1.0	0.0	3.3 ± 1.5	9.9 ± 1.1	2.9 ± 1.1
Qwen2.5-7B-Instruct																
Vanilla	28.6	30.8	11.1	0.0	0.0	0.0	12.1	41.7	27.8	0.0	4.3	23.8	0.0	15.7	2.4	0.2
GRPO	92.6	75.4	84.4	77.5	65.6	70.0	79.0 ± 1.3	73.3	95.6	78.7	79.1	86.7	68.2	80.0 ± 1.7	83.0 ± 0.7	73.3 ± 0.6
RLOO	93.7	83.1	76.3	87.5	67.2	48.3	76.1 ± 1.6	76.7	87.8	72.3	81.7	86.7	36.5	74.5 ± 0.8	86.2 ± 0.7	73.0 ± 0.5
HGPO	98.9	87.7	**99.3**	**93.8**	82.4	91.7	93.1 ± 1.6	82.5	98.9	92.3	91.3	96.2	**98.8**	**92.7 ± 1.3**	86.0 ± 0.6	73.1 ± 0.9
GiGPO	**100.0**	89.2	97.0	**93.8**	88.0	90.8	94.0 ± 1.2	79.2	96.7	87.7	93.9	89.5	94.1	89.6 ± 1.8	82.0 ± 0.5	72.1 ± 1.0
GraphGPO	98.9	**96.9**	**99.3**	**93.8**	91.2	**95.0**	**96.1 ± 1.1**	79.2	**100.0**	**92.9**	89.6	95.2	97.6	91.8 ± 1.2	86.3 ± 0.5	75.4 ± 1.0
SDAR	97.1	89.2	88.1	86.2	66.4	71.7	83.6 ± 1.8	**87.5**	86.7	82.6	75.7	61.9	81.2	79.4 ± 2.9	81.8 ± 0.5	67.7 ± 1.1
AgentOPSD	96.6	70.8	88.1	70.0	79.2	64.2	80.9 ± 2.0	80.0	68.9	88.4	66.1	89.5	51.8	76.0 ± 1.1	86.1 ± 0.5	71.0 ± 0.2
Agent-G^2	97.7	72.3	98.5	91.2	88.0	89.2	91.6 ± 1.8	85.0	74.4	91.6	91.3	93.3	96.5	89.0 ± 1.3	86.4 ± 1.0	72.8 ± 1.0
CPAD	**100.0**	92.3	93.3	**93.8**	**94.4**	93.3	95.1 ± 0.8	79.2	98.9	83.9	**94.8**	**100.0**	94.1	90.7 ± 0.9	**88.9 ± 0.2**	**79.2 ± 0.4**
CPAD+	63.4	47.7	43.7	66.2	66.4	43.3	55.6 ± 2.1	54.2	30.0	50.3	61.7	66.7	20.0	49.0 ± 0.9	78.2 ± 0.6	67.9 ± 0.7
CPAD−	22.3	10.8	5.2	2.5	5.6	0.8	9.0 ± 2.6	16.7	17.8	1.9	2.6	3.8	0.0	6.9 ± 1.1	20.8 ± 1.9	8.0 ± 1.1

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks. ALFWorld is a text-based embodied benchmark containing 3,827 household tasks across six families (Shridhar et al., 2021): Pick & Place (**Pick**), Examine in Light (**Look**), Clean & Place (**Clean**), Heat & Place (**Heat**), Cool & Place (**Cool**), and Pick Two & Place (**Pick2**). WebShop is a simulated e-commerce environment with 1.18 million products and 12,087 instructions requiring agents to find and purchase matching products (Yao et al., 2023).

Models and baselines. We use Qwen2.5-Instruct models at 1.5B, 3B, and 7B scales. RLOO (Ahmadian et al., 2024) and GRPO (Shao et al., 2024) provide outcome-based group-relative baselines. HGPO (He et al., 2026), GiGPO (Feng et al., 2025), and GraphGPO (Cheng et al., 2026) provide history-conditioned, state-level, and graph-structured credit-assignment baselines, respectively. SDAR (Lu et al., 2026) and AgentOPSD (Wang et al., 2026a) provide on-policy distillation baselines. Agent-G^2 (Wang et al., 2026b) samples expert-prefix guidance depth from task-adaptive Gaussian distributions estimated from online rollouts. We evaluate CPAD together with its positive-only (CPAD+) and negative-only (CPAD−) branch variants. Within each benchmark and backbone,

the comparison protocol uses a common training-task pool, rollout budget, interaction horizon, and checkpoint cadence. Appendix H lists the common hyperparameters and benchmark–model settings.

Matched full-set evaluation. Every five policy updates, we evaluate each checkpoint on the task pools in Table 1 using the same decoding protocol, interaction limits, and scoring rules.

4.2 Main Results

Table 1 reports full-set validation, with CPAD reaching 95.6% ALFWorld seen success and 73.3% WebShop success at 3B. Across ALFWorld seen and WebShop at all three scales, CPAD improves final success and area under the 150-update validation curve (AUC) over matched GRPO; the largest final-success gains are 24.4 and 11.4 percentage points on the two benchmarks, respectively.

4.3 Learning Dynamics and One-Sided Privileged Signals

Figure 3 jointly compares validation success and the logged reference KL loss, which measures policy drift from the reference model, for GRPO, paired CPAD, CPAD+, and CPAD−. Paired CPAD achieves higher final success than both one-sided variants in five of six settings and lower mean reference KL in all six. On WebShop 7B, CPAD+ and CPAD− reach 67.9% and 8.0% final success, respectively, compared with 79.2% for paired CPAD; their mean reference KL losses are 0.614 and 0.389 versus 0.219. On ALFWorld 7B, CPAD+ and CPAD− reach 55.6% and 9.0% final success, respectively, compared with 95.1% for paired CPAD; their mean reference KL losses are 0.909 and 0.644 versus 0.300. On WebShop 3B, CPAD+ reaches 73.4% final success and higher AUC than paired CPAD, while paired CPAD maintains lower mean reference KL. These results support combining preferred- and inferior-action guidance for strong task performance with smaller reference-policy drift.

CPAD selects 8.0–25.2% of decisions on average across the six settings (Appendix C). Appendices B and D analyze correction strength and eleven additional parameter variants. On WebShop 1.5B, averaging or shuffling the clipped corrections within each response reduces validation-success AUC from 48.90% to 39.46% and 39.43%, respectively, with the selector and paired teachers unchanged. These results support preserving both within-response variation and token alignment (Appendix E).

4.4 Interaction and Training Efficiency

At 150 updates with the same rollout budget, CPAD uses 24.1–48.7% fewer training interactions than GRPO and achieves higher final success in all six settings (Figure 4(a)). It also achieves higher success AUC than HGPO, GiGPO, and GraphGPO within a common training-time budget across all six settings (Figure 4(b)). Appendix F provides learning curves, costs of reaching target success rates, and timing breakdowns.

5 Related Work

5.1 Credit Assignment in Agentic Reinforcement Learning

Outcome-based methods such as RLOO and GRPO estimate relative advantages from sampled groups without a learned critic (Ahmadian et al., 2024; Shao et al., 2024). In multi-turn interaction, however, terminal rewards cannot distinguish useful decisions in failed trajectories from harmful ones in successful trajectories. GiGPO adds step-level advantages by grouping actions at repeated anchor states (Feng et al., 2025). HGPO notes that equal current states can have different histories and combines history-conditioned groups to balance bias and variance (He et al., 2026). GraphGPO merges rollout transitions into a graph and rewards progress toward empirically reachable goals (Cheng et al., 2026). Implicit step-reward methods instead learn process rewards from trajectory preferences and combine step- and episode-level advantages (Liu et al., 2025).

CPAD uses related rollout structure to select and condition privileged supervision. It converts supported action preferences at matched decision states into token-level OPD corrections, retaining the outcome advantage as the global anchor.

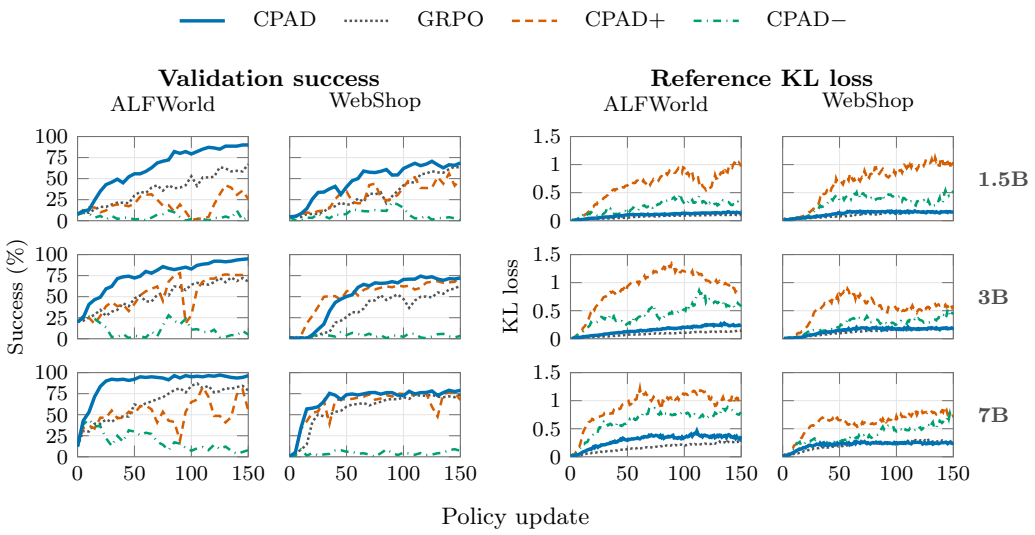

Figure 3: CPAD improves validation learning over matched GRPO while the paired variants exhibit lower mean reference KL than their one-sided counterparts across all six settings. Rows denote Qwen2.5 model scales; the left and right column groups report validation success and reference KL loss, respectively, on ALFWorld and WebShop. ALFWorld success is measured on the 140-task `val_seen` set. All displayed curves cover updates through 150.

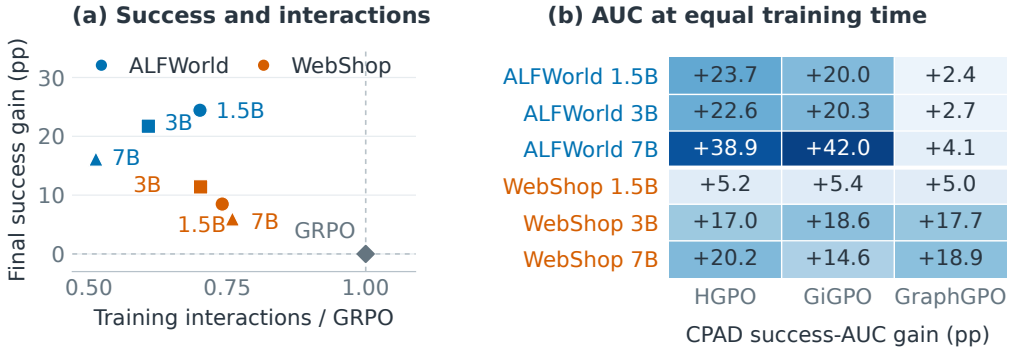

Figure 4: Interaction and training efficiency across six settings. (a) CPAD's training-interaction ratio and final-success gain relative to GRPO at 150 updates. (b) Success-AUC gains over structured credit-assignment baselines within each setting's common training-time budget. ALFWorld uses `val_seen`. Interactions sum training-trajectory lengths; training time includes privileged scoring and excludes evaluation and checkpointing.

5.2 On-Policy Distillation for Agentic Reinforcement Learning

Agentic OPD methods differ in the source of privileged information, the unit of supervision, and its combination with RL. SDAR retrieves training-only skills and gates auxiliary self-distillation, strengthening teacher-endorsed updates while attenuating unreliable negative gaps (Lu et al., 2026). OPID extracts episode-level and critical-step skills from completed on-policy trajectories, routing them to decisions before forming token-level advantages (Yang et al., 2026b). SEED uses the current model to collect trajectories and analyze them into skills, allowing hindsight supervision to evolve with the policy (Wu et al., 2026a).

StepOPSD re-scores action-centered spans with successful-peer hindsight and applies bounded, sign-preserving step-level advantage shaping (Zhang et al., 2026). AgentOPSD aggregates token gaps into turn-level evidence, recursively revises a trajectory-success belief, and uses its marginal

changes to redistribute outcome credit (Wang et al., 2026a). These methods show why a raw teacher–student gap requires interpretation in sequential settings. CPAD instead conditions on a preferred–inferior action pair from matched decisions. The same local evidence determines both activation and contrast, without retrieved skills or analyzer-generated hindsight; its additive token correction can change the advantage sign.

5.3 Reliability and Contrast in Privileged Supervision

RLSD identifies privileged-information leakage and long-term OPSD instability, using outcome rewards to anchor update directions (Yang et al., 2026a). SDAR reports instability in naive GRPO+OPSD and amplified RLSD-style updates, motivating gated auxiliary supervision (Lu et al., 2026). CRPO addresses unreliable position-level supervision and rising KL with entropy-based selection and group-wise contrast (Wu et al., 2026b). RLCSD identifies style drift from one-sided positive contexts in mathematical and logical reasoning and contrasts correct- and incorrect-hint signals (Pan et al., 2026).

CPAD shares this concern with reliability and contrast. Its distinction is that matched local action outcomes jointly determine where supervision acts and which two action contexts define its direction.

6 Limitations

CPAD relies on comparable decision contexts and supported action preferences within on-policy rollouts. Such comparisons may be less frequent in environments with highly diverse observations or few successful trajectories, reducing the opportunities for additional local supervision. Studying richer state representations while preserving decision comparability is a natural direction for extending its applicability.

Our evaluation covers ALFWorld and WebShop with Qwen2.5 models from 1.5B to 7B. Extending CPAD to multimodal environments and less structured action spaces remains an open direction, requiring further study of both decision matching and the construction of paired action guidance.

7 Conclusion

We introduced CPAD for on-policy distillation in long-horizon agents. It uses comparisons of actions based on rollout outcomes at matched decisions to select where distillation acts and construct paired privileged guidance, producing token-level corrections to the outcome advantage. Across ALFWorld and WebShop at three Qwen2.5 model scales, CPAD improves validation AUC over GRPO in all six settings, achieves performance on par with the state of the art, and uses fewer training interactions at the same update budget. Branch ablations support combining preferred and inferior action guidance.

AI USE STATEMENT

We used generative AI tools to assist with manuscript drafting and editing, code development, and the creation of illustrative icons for the method overview. All AI-assisted material was reviewed by the authors. The authors take full responsibility for the final manuscript, code, and reported results.

ETHICS STATEMENT

This work studies agent training using the existing ALFWorld and WebShop benchmarks. All reported agent interactions take place in simulated environments. The study does not involve recruiting human participants or collecting new personal data.

REPRODUCIBILITY STATEMENT

Section 3 describes the method, decision selection procedure, and training objective. Appendix G provides the interaction and privileged prompt templates, and Appendix H reports the experimental settings. Section 4 specifies the benchmarks and evaluation protocol. To support reproducibility, we will publicly release the complete implementation, training and evaluation scripts, experiment configurations, and the experimental records and analysis scripts underlying the reported tables and figures. The release will document software and model versions, random seeds, and environment setup, and provide instructions for obtaining the benchmark resources.

REFERENCES

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

Xin Cheng, Shuo He, Lang Feng, HaiYang Xu, Ming Yan, Lei Feng, and Bo An. Beyond trajectory-level attribution: Graph-based credit assignment for agentic reinforcement learning, 2026.

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

Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-groups policy optimization for long-horizon agentic tasks, 2026.

Ahmed Heakl, Abdelrahman M. Shaker, Youssef Mohamed, Rania Elbadry, Omar Fetouh, Fahad Shahbaz Khan, and Salman Khan. CEPO: RLVR self-distillation using contrastive evidence policy optimization, 2026.

Xiaoqian Liu, Ke Wang, Yuchuan Wu, Fei Huang, Yongbin Li, Junge Zhang, and Jianbin Jiao. Agentic reinforcement learning with implicit step rewards, 2025.

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, 2026.

Leyi Pan, Shuchang Tao, Yunpeng Zhai, Lingzhe Zhang, Zhaoyang Liu, Bolin Ding, Aiwei Liu, and Lijie Wen. RLCSD: Reinforcement learning with contrastive on-policy self-distillation, 2026.

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

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, 2024.

Guobin Shen, Lei Huang, Xiang Cheng, Chenxiao Zhao, Jindong Li, Dongcheng Zhao, and Xing Yu. From generic correlation to input-specific credit in on-policy self distillation, 2026.

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

Zi-Han Wang, Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Jie Wu, Zhengzhou Cai, Yueqing Sun, Ziang Ye, Linji Hao, Qi Gu, Xunliang Cai, Yongliang Shen, and Yujiu Yang. AgentOPSD: Recursive self-distillation for agentic reinforcement learning, 2026a.

Zixuan Wang, Yanrui Miao, Zhengxi Lu, Teng Pan, Yiwen Qiu, Hongxing Li, Peng Qiu, Ruiqing Zhang, and Yongliang Shen. Agent-G^2: Gaussian guidance for agentic reinforcement learning, 2026b.

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, 2026a.

Xingjian Wu, Junlin Liu, Xingchen Liu, Xuhang Zhu, Jianing Wang, Linsen Guo, Xiaoyu Li, Xuezhi Cao, and Xunliang Cai. Contrastive reinforced policy optimization via privileged self-distillation, 2026b.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled RLVR, 2026a.

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, 2026b.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents, 2023.

Yanfei Zhang, Xu Lin, and Chenglin Wu. StepOPSD: Step-aware online preference self-distillation for agent reinforcement learning, 2026.

A CPAD TRAINING PROCEDURE

Algorithm 1 summarizes CPAD training. ROLLOUT samples trajectories under ordinary contexts $h^0_{i,t}$, and GROUPSTATS computes the decision-occurrence mean and sample standard deviation in Equation 14. SELECTPAIR applies the support and continuation-quality rules in Section 3.1, returning $(\perp, \perp)$ when it abstains. Per-task assignments range over $(i,t) \in \mathcal{B}_x$; token-indexed assignments range over $k \in \mathcal{K}_{i,t}$.

Algorithm 1 Contrastive Privileged Action Distillation

Input: Policy π_θ, training tasks, rollout count N, correction weight λ, bound c_Δ, selector settings

for each policy update **do**
  $\bar{\theta} \leftarrow \theta$
  for each sampled task x **do**
    $\{(\tau_i, R_i)\}_{i=1}^N \leftarrow \text{ROLLOUT}(\pi_{\bar{\theta}}, x, N)$
    $\mathcal{B}_x \leftarrow \{(i,t) : 1 \le i \le N,\ 1 \le t \le T_i\}$
    $\widetilde{R}_{i,t} \leftarrow c_R R_i - c_{\text{inv}} \mathbf{1}\{a_{i,t} \text{ is invalid}\}$
    $(\mu_x, \sigma_x) \leftarrow \text{GROUPSTATS}(\{\widetilde{R}_{i,t}\}_{(i,t)\in\mathcal{B}_x})$ ▷ Equation 14
    $A^{\text{out}}_{i,t} \leftarrow (\widetilde{R}_{i,t} - \mu_x)/(\sigma_x + \epsilon)$
    $g_{i,t} \leftarrow 0; \quad \widetilde{\Delta}_{i,t,k} \leftarrow 0$
    $z_{i,t} \leftarrow \text{key}(x, \xi_{i,t}); \quad \mathcal{G}_z \leftarrow \{(i,t) \in \mathcal{B}_x : z_{i,t} = z\}$
    for each matched group $\mathcal{G}_z$ **do**
      $\mathcal{I}_z(a) \leftarrow \{i : \exists t,\ (i,t) \in \mathcal{G}_z,\ a_{i,t} = a\}$
      $\widehat{r}_z(a) \leftarrow \sum_{i\in\mathcal{I}_z(a)} R_i / |\mathcal{I}_z(a)|$ ▷ Each observed action a
      $(a_z^+, a_z^-) \leftarrow \text{SELECTPAIR}(\mathcal{G}_z)$ ▷ Section 3.1
      if $a_z^+ \neq \perp$ **then**
        for all $(i,t) \in \mathcal{G}_z$ **do**
          $g_{i,t} \leftarrow 1; \quad h^{\pm}_{i,t} \leftarrow \mathcal{P}(h^0_{i,t}, a_z^{\pm})$
          $\ell^{\pm}_{i,t,k} \leftarrow \log \pi_{\bar{\theta}}(y_{i,t,k} \mid h^{\pm}_{i,t}, y_{i,t,<k})$
          $\Delta_{i,t,k} \leftarrow \ell^+_{i,t,k} - \ell^-_{i,t,k}$
          $\widetilde{\Delta}_{i,t,k} \leftarrow g_{i,t}\ \text{sg}[\text{clip}(\Delta_{i,t,k}, -c_\Delta, c_\Delta)]$
        end for
      end if
    end for
    $A^{\text{CPAD}}_{i,t,k} \leftarrow A^{\text{out}}_{i,t} + \lambda \widetilde{\Delta}_{i,t,k}$
  end for
  Update θ with $\mathcal{L}_{\text{actor}}(A^{\text{CPAD}})$ on the collected batch under $h^0_{i,t}$
end for
return π_θ

Outcome advantage implementation. For task x, let $\mathcal{B}_x = \{(i,t) : 1 \le i \le N,\ 1 \le t \le T_i\}$ and $M_x = |\mathcal{B}_x| = \sum_{i=1}^N T_i$. For $M_x > 1$, Algorithm 1 uses the adjusted rewards and normalization statistics

$$\widetilde{R}_{i,t} = c_R R_i - c_{\text{inv}}\, \mathbf{1}\{a_{i,t} \text{ is invalid}\}, \tag{13}$$

$$\mu_x = \frac{1}{M_x} \sum_{(j,u)\in\mathcal{B}_x} \widetilde{R}_{j,u}, \qquad \sigma_x^2 = \frac{1}{M_x - 1} \sum_{(j,u)\in\mathcal{B}_x} (\widetilde{R}_{j,u} - \mu_x)^2. \tag{14}$$

Each decision contributes once, so a trajectory with T_i decisions contributes T_i rewards to the mean and sample variance. Appendix H specifies the reward coefficients and normalization constant.

B LAMBDA SENSITIVITY

Figure 5 examines the correction strength $\lambda \in \{0.25, 0.5, 1, 2\}$ on WebShop with Qwen2.5-1.5B-Instruct. Increasing λ from 0.5 to 2 reduces normalized validation-success AUC from 46.7% to 39.7%, while the mean reference KL loss rises from 0.122 to 0.221. The default $\lambda = 1$ achieves the highest AUC in this sweep (48.9%).

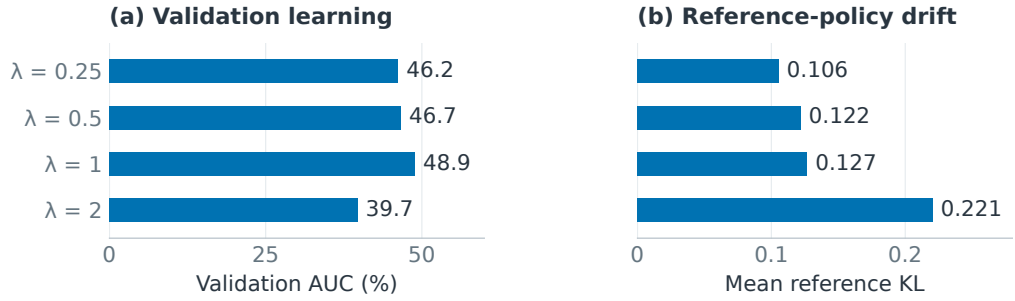

Figure 5: Correction-strength sensitivity on WebShop, Qwen2.5-1.5B. Aligned rows share λ. (a) Trapezoidal area under validation success over updates 0–150, divided by 150 and expressed as a percentage. (b) Mean logged reference KL loss over updates 1–150.

C DECISION SELECTION COVERAGE

To quantify how selectively CPAD intervenes during training, we measure the fraction of decision occurrences selected for privileged supervision in each rollout batch. Both the numerator and denominator include only occurrences with at least one valid response token. Table 2 reports the arithmetic mean of this fraction over updates 1–150 for the six CPAD runs used in Table 1, with equal weight for each update.

Table 2: Mean decision selection coverage (%) over 150 policy updates.

Benchmark	1.5B	3B	7B
ALFWorld	25.2	24.0	25.1
WebShop	14.9	8.0	8.9

Mean coverage ranges from 24.0% to 25.2% on ALFWorld and from 8.0% to 14.9% on WebShop. These observations quantify CPAD's sparsity across decisions.

D SELECTION AND CONTRAST SENSITIVITY

We supplement the correction-strength study with eleven configurations on WebShop using Qwen2.5-1.5B-Instruct. Each run uses 150 policy updates, eight rollouts per task, and the same 500-goal evaluation pool. Table 3 reports both endpoint performance and learning-curve AUC, together with decision coverage and reference KL. Figure 6 shows the full validation trajectories.

Parameters and controls. The default is $h = 0$, $(m_+, m_-) = (1, 1)$, $c_\Delta = 5$, and $G_{\min} = 2$, with $\lambda = 1$ and unit weights for both guidance branches. Here h is the history length in the *matching key*; the ordinary agent context retains up to two observation–action pairs in every run. The thresholds m_+ and m_- count distinct trajectories supporting the preferred and alternative actions, respectively, using $\mathcal{I}_z(a)$ in Equation 6. The negative threshold applies to both lower-success and continuation-dominated alternatives, counting support from successful and failed trajectories. Preferred actions must have at least one successful supporting trajectory in every setting. Finally, $G_{\min}$ counts distinct trajectories in the whole decision group.

Table 3: WebShop 1.5B parameter sensitivity. Score and Success use update 150. AUC is the trapezoidal validation-success integral over updates 0–150 divided by 150. Coverage and reference KL are arithmetic means over updates 1–150. All columns except KL are percentages.

Setting	Score	Success	AUC	Coverage	Ref. KL
GRPO	76.27	62.8	31.54	—	0.105
CPAD default	85.41	68.6	48.90	14.94	0.127
$h = 1$	81.55	65.0	46.47	9.26	0.136
$h = 2$	85.14	72.6	44.65	7.22	0.113
$(m_+, m_-) = (2, 1)$	83.08	64.0	43.91	7.26	0.101
$(m_+, m_-) = (1, 2)$	85.27	71.8	46.13	11.94	0.125
$(m_+, m_-) = (2, 2)$	84.88	71.4	47.17	4.69	0.112
$c_\Delta = 0.5$	80.11	62.8	42.53	23.94	0.152
$c_\Delta = 1$	85.12	72.6	49.93	12.91	0.122
$c_\Delta = 2$	86.71	72.6	51.16	10.83	0.119
$c_\Delta = 10$	85.86	71.4	52.04	8.15	0.119
$G_{\min} = 3$	82.93	71.2	51.23	8.92	0.126
$G_{\min} = 4$	85.12	70.6	50.46	9.64	0.110

Matching history trades coverage for specificity. With $h = 1$ and $h = 2$, mean coverage is 9.26% and 7.22%, respectively. The longer-history run reaches 72.6% final success, but its AUC is 44.65%, below 46.47% for $h = 1$. More specific matching reduces coverage and exposes a trade-off between the area under the learning curve and final success.

Positive and negative support have different effects. Requiring two preferred-action supports alone yields 64.0% final success, versus 71.8% when requiring two alternative-action supports alone. Requiring both yields 71.4%, despite selecting only 4.69% of decisions on average. The asymmetric outcomes motivate exposing separate support thresholds. Useful supervision persists even when selection covers fewer than 5% of decisions.

Contrast clipping has a broad effective range. The settings $c_\Delta \in \{1, 2, 10\}$ reach 71.4–72.6% final success; $c_\Delta = 10$ has the highest AUC in the sweep (52.04%), while $c_\Delta = 2$ has the highest partial-credit Score (86.71%). The tighter bound 0.5 yields 62.8% final success and the highest mean reference KL (0.152), combining weaker task performance with greater reference-policy drift.

Larger decision groups remain effective. Requiring three or four trajectories per group yields 71.2% and 70.6% final success, with AUCs of 51.23% and 50.46%. Mean coverage is 8.92% and 9.64%, respectively. Training-average coverage reflects both the group threshold and the rollout distribution induced by the learned policy.



Figure 6: Validation success for all eleven parameter settings, grouped by the changed parameter. Curves are unsmoothed and share updates 0–150. Black dashed and gray dotted curves repeat the default CPAD and GRPO references. In (b), pairs denote (m_+, m_-); in (c), labels denote c_Δ; in (d), G denotes $G_{\min}$.

Performance across settings. All eleven variants have higher AUC than the GRPO reference; ten also have higher final success and the strongest-clipping variant ties it. These gains span the tested matching histories, action-support thresholds, contrast bounds, and group sizes.

E TOKEN-LEVEL CORRECTION ABLATIONS

We examine whether CPAD benefits from within-response variation, token alignment, and informative correction values on WebShop with Qwen2.5-1.5B-Instruct. All settings use 150 policy updates, eight rollouts per task, and the same 500-goal evaluation pool.

Controls. All three controls retain the decision selector, paired privileged contexts, outcome advantage, $\lambda = 1$, and $c_\Delta = 5$. *Response mean* replaces the clipped correction $\widetilde{\Delta}_{i,t,k}$ within each selected response with its valid-token mean. *Token shuffle* permutes these values within each response, preserving their distribution while breaking token alignment. *Random correction* replaces the correction with $g_{i,t}\,\mathrm{clip}(\sigma\epsilon_{i,t,k}, -c_\Delta, c_\Delta)$, where $\epsilon_{i,t,k}$ are independent standard Gaussian draws. We fix $\sigma \approx 0.40577$ before Random training using the token-count-weighted RMS of default CPAD's clipped corrections over updates 1–150. Teacher scoring is retained but its values are discarded. Noise uses a separate generator (base seed 0 plus the update index), without response centering or online rescaling.

Table 4: Token-level correction ablations on WebShop 1.5B. Score and Success use the training-time evaluation at update 150; AUC integrates validation success over updates 0–150 and divides by 150. Coverage and reference KL are means over updates 1–150. All columns except KL are percentages.

Setting	Score	Success	AUC	Coverage	Ref. KL
GRPO	76.27	62.8	31.54	—	0.105
CPAD	85.41	68.6	48.90	14.94	0.127
Response mean	83.22	65.0	39.46	26.40	0.094
Token shuffle	60.69	50.4	39.43	32.30	0.120
Random correction	79.82	61.4	36.28	34.62	0.107

Token-level information and alignment matter. CPAD's AUC is 48.90%, versus 39.46% for response mean and 39.43% for token shuffle; final success is 68.6%, 65.0%, and 50.4%, respectively (Table 4). The loss from averaging supports retaining within-response variation; the loss from shuffling supports aligning it with the tokens being optimized. Random correction has higher AUC than GRPO (36.28% versus 31.54%) but lower final success (61.4% versus 62.8%). It does not recover CPAD's gains, supporting the value of informative, aligned token-level corrections in this setting (Figure 7).

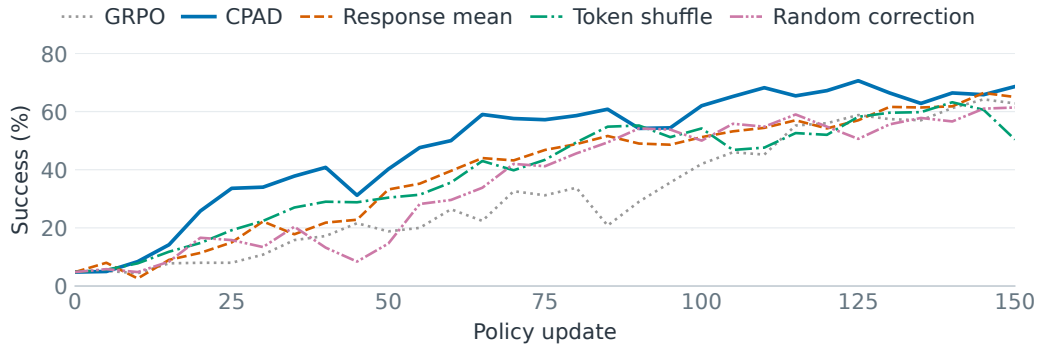

Figure 7: Validation success on WebShop 1.5B for GRPO, CPAD, and all three correction controls. Curves show every five-update evaluation without smoothing.

Coverage alone does not explain the gains. Response mean, token shuffle, and random correction have mean decision coverage of 26.40%, 32.30%, and 34.62%, respectively, versus 14.94% for CPAD, yet lower AUC. These measurements use each learned policy's own rollout distribution with unchanged selector rules. Broader observed coverage therefore does not recover CPAD's gains.

F Learning Efficiency against Structured Credit Assignment

We compare CPAD with HGPO, GiGPO, and GraphGPO across ALFWorld and WebShop at all three model scales. Table 5 summarizes learning across all six settings; Figures 8 and 9 show the 3B and 7B curves in both environments. All methods train for 150 updates with 16 tasks per update and eight rollouts per task. Success is evaluated every five updates on ALFWorld `val_seen` and the fixed WebShop validation goals.

Budgets and area under the learning curve. Cumulative interactions sum the lengths of all training trajectories. Training time sums the recorded update durations, including privileged re-scoring and excluding evaluation and checkpoint saving. For success curve $S(b)$ against budget b, we report normalized area

$$\mathrm{AUC}_B = \frac{100}{B} \int_0^B S(b)\,\mathrm{d}b.$$

Update-based AUC uses $B = 150$. For interaction- and time-based comparisons, B is the smallest completed budget among the four methods in each setting. We integrate validation checkpoints by the trapezoidal rule, linearly interpolating the common-budget boundary. The vertical dotted lines in the learning curves mark these common budgets.

Table 5: Normalized validation-success AUC (%). Each block uses a shared budget within each setting: 150 updates or the common cumulative training time. Bold marks the highest value in each block.

	150 updates				Common training time			
Setting	HGPO	GiGPO	GraphGPO	CPAD	HGPO	GiGPO	GraphGPO	CPAD
ALFWorld 1.5B	53.12	53.13	**66.70**	63.89	35.08	38.73	56.31	**58.75**
ALFWorld 3B	65.42	66.31	**77.12**	75.63	47.41	49.75	67.27	**70.01**
ALFWorld 7B	77.20	77.90	84.92	**88.55**	43.57	40.43	78.33	**82.43**
WebShop 1.5B	45.13	45.50	46.07	**48.90**	37.43	37.30	37.70	**42.65**
WebShop 3B	42.36	39.97	42.60	**51.12**	24.85	23.19	24.10	**41.83**
WebShop 7B	55.61	57.80	56.34	**68.20**	44.09	49.64	45.39	**64.28**

Learning efficiency. CPAD exceeds HGPO and GiGPO in update-based AUC in all six settings, and GraphGPO in four. Under a common training-time budget, CPAD achieves higher AUC than all three baselines in every setting. On WebShop 3B and 7B, its AUC gains over GraphGPO are 17.73 and 18.89 percentage points, respectively. On ALFWorld 1.5B and 3B, GraphGPO has higher update-based AUC, while CPAD has higher AUC within the common training time. The interaction curves also show earlier success gains on WebShop and ALFWorld 7B.

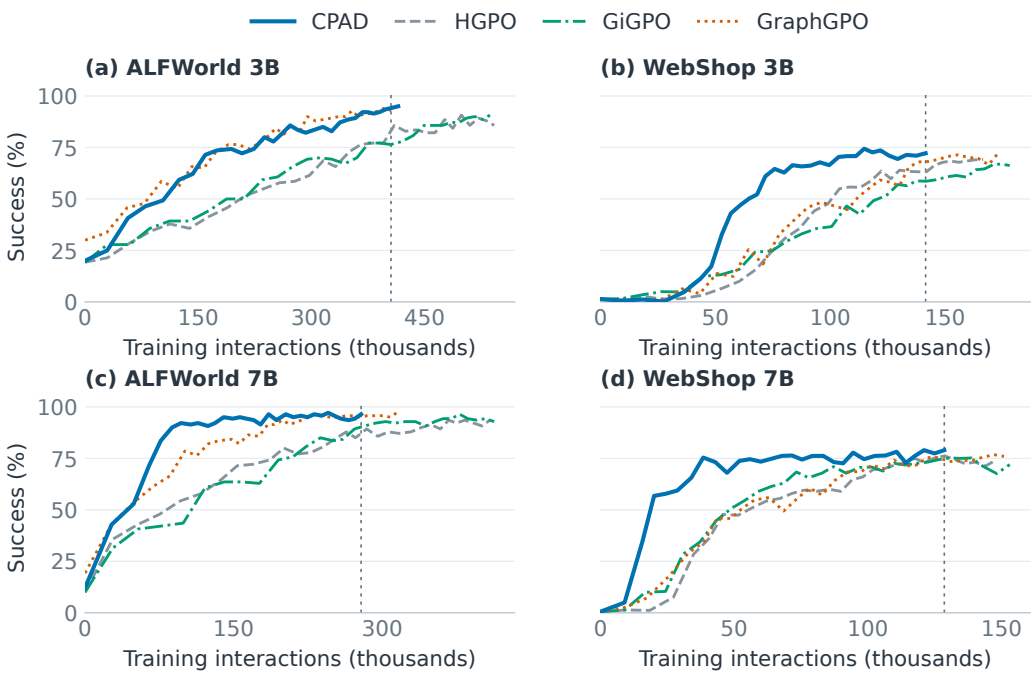

Figure 8: Validation success versus cumulative training interactions for CPAD and three structured credit-assignment baselines. All curves extend through update 150. ALFWorld uses `val_seen`; vertical dotted lines mark the interaction budgets shared by all four methods in each panel.

Cost of reaching target success. We use a common target of 90% for ALFWorld and 70% for WebShop, requiring success to meet the target at two consecutive evaluations. Costs are measured at the second checkpoint of the first qualifying pair. Table 6 reports the resulting interactions and training hours. On WebShop 7B, CPAD reaches the target with 58–62% fewer interactions and 57–65% less training time than the three baselines. On ALFWorld 7B, it uses 49.3% fewer interactions and 45.9% less time than GraphGPO. On ALFWorld 3B, GraphGPO reaches the target with fewer interactions, while CPAD reaches it in less training time.

Table 6: Costs of reaching 90% success on ALFWorld and 70% on WebShop at two consecutive evaluations. I: cumulative interactions in thousands; T: cumulative training hours. Bold marks the lowest attained cost for each metric; a dash indicates that the target condition was not met within 150 updates.

	HGPO		GiGPO		GraphGPO		CPAD	
Setting	I	T	I	T	I	T	I	T
ALFWorld 3B	—	—	—	—	**344.6**	12.93	374.8	**11.17**
ALFWorld 7B	348.8	46.11	293.3	37.42	190.9	10.29	**96.8**	**5.56**
WebShop 3B	—	—	—	—	155.4	9.91	**107.4**	**6.16**
WebShop 7B	113.4	12.50	101.4	10.46	113.8	12.80	**42.9**	**4.49**

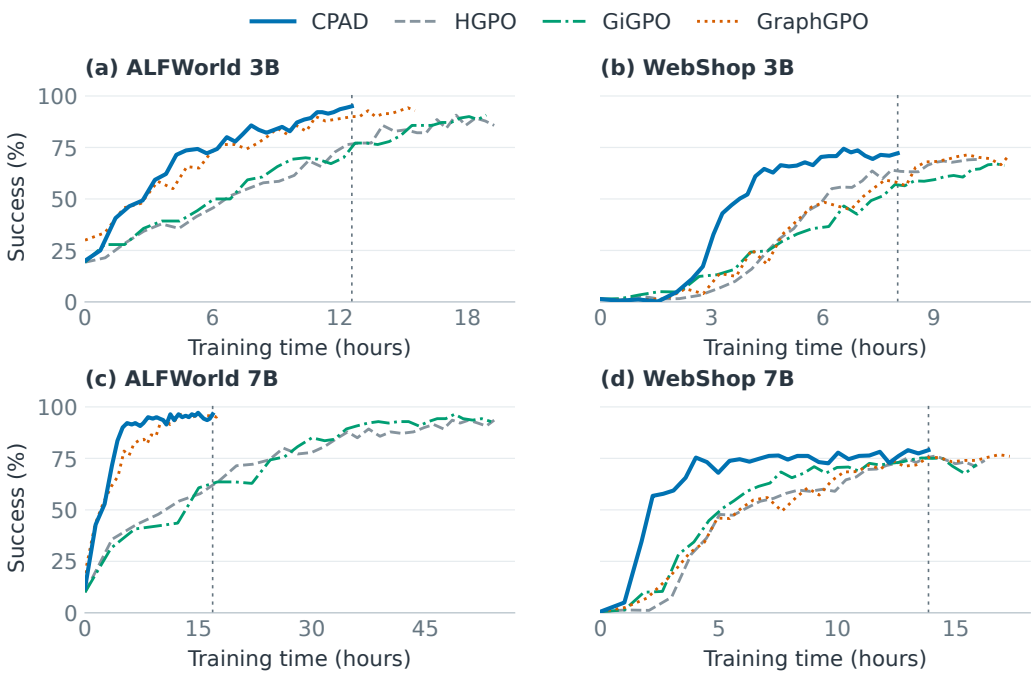

Figure 9: Validation success versus cumulative training time, including privileged re-scoring and excluding evaluation and checkpoint saving. The settings and methods match Figure 8. Vertical dotted lines mark the common time budgets used for AUC.

Privileged scoring cost. The two teacher-forced scoring passes account for 1.89–6.49% of CPAD's training time across the six settings. Figure 10 separates this cost from the rest of training and from evaluation/checkpointing in the comparison with GRPO. Including privileged scoring, CPAD's training time is lower than GRPO, HGPO, GiGPO, and GraphGPO in all six settings. Relative to GraphGPO, total training time decreases by 5.0–27.4%.

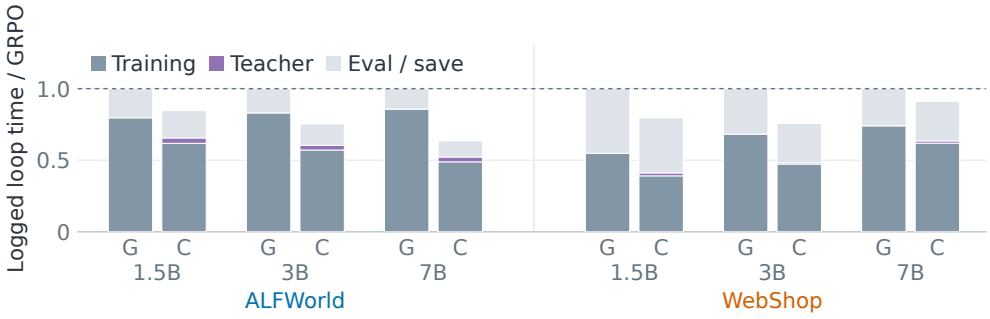

Figure 10: Recorded loop-time breakdown at 150 updates. G/C denote GRPO/CPAD; each pair is normalized by GRPO's total loop time. Stacks separate other training, privileged teacher scoring, and evaluation/checkpointing. Timers exclude initial evaluation, startup, and downtime.

G PROMPT TEMPLATES

The following templates specify the user-message content supplied to the Qwen2.5-Instruct tokenizer's chat template, with an assistant-generation prefix. Braced fields are replaced with the current task and environment information. Both benchmarks use an initial-decision template and a subsequent-decision template containing up to two preceding observation–action pairs. Each history entry has the form `[Observation j: 'observation', Action j: 'action']`; `history_length` is the number of available entries, up to two. Line wrapping and typographic hyphens are normalized for display.

G.1 ALFWORLD INTERACTION PROMPTS

The initial observation includes the task description. Admissible actions are listed in single quotes, one per line, with the environment's `help` action excluded.

ALFWorld: initial decision

```
You are an expert agent operating in the ALFRED Embodied Environment.
Your current observation is: {current_observation}
Your admissible actions of the current situation are:
[{admissible_actions}].

Now it's your turn to take an action.
You should first reason step-by-step about the current situation. This
reasoning process MUST be enclosed within <think> </think> tags.
Once you've finished your reasoning, you should choose an admissible
action for current step and present it within <action> </action> tags.
```

ALFWorld: subsequent decisions

```
You are an expert agent operating in the ALFRED Embodied Environment.
Your task is to: {task_description}
Prior to this step, you have already taken {step_count} step(s). Below
are the most recent {history_length} observations and the corresponding
actions you took: {action_history}
You are now at step {current_step} and your current observation is:
{current_observation}
Your admissible actions of the current situation are:
[{admissible_actions}].

Now it's your turn to take an action.
You should first reason step-by-step about the current situation. This
reasoning process MUST be enclosed within <think> </think> tags.
Once you've finished your reasoning, you should choose an admissible
action for current step and present it within <action> </action> tags.
```

G.2 WEBSHOP INTERACTION PROMPTS

Available actions are formatted as `search[<your query>]` when a search bar is present and `click[<item>]` for each clickable item. Each action is single-quoted, followed by a comma, on its own line. The initial-decision template is also used when the history-augmented prompt exceeds 13,000 characters.

WebShop: initial decision

```
You are an expert autonomous agent operating in the WebShop e-commerce
environment.
Your task is to: {task_description}.
Your current observation is: {current_observation}.
Your admissible actions of the current situation are:
[
{available_actions}
].

Now it's your turn to take one action for the current step.
```

```
You should first reason step-by-step about the current situation, then
think carefully which admissible action best advances the shopping
goal. This reasoning process MUST be enclosed within <think> </think>
tags.
Once you've finished your reasoning, you should choose an admissible
action for current step and present it within <action> </action> tags.
```

WebShop: subsequent decisions

```
You are an expert autonomous agent operating in the WebShop e-commerce
environment.
Your task is to: {task_description}.
Prior to this step, you have already taken {step_count} step(s). Below
are the most recent {history_length} observations and the corresponding
actions you took: {action_history}
You are now at step {current_step} and your current observation is:
{current_observation}.
Your admissible actions of the current situation are:
[
{available_actions}
].

Now it's your turn to take one action for the current step.
You should first reason step-by-step about the current situation, then
think carefully which admissible action best advances the shopping
goal. This reasoning process MUST be enclosed within <think> </think>
tags.
Once you've finished your reasoning, you should choose an admissible
action for current step and present it within <action> </action> tags.
```

G.3 Privileged Action Guidance

For either ordinary template above, CPAD inserts the corresponding guidance sentence immediately after the admissible-action list and before the line beginning `Now it's your turn`. The rest of the prompt is retained.

ALFWorld: inserted guidance

```
Your teacher has provided the following guidance for the current
decision: taking the action '{action}' would be better.
```

WebShop: inserted guidance

```
Your teacher has provided the following guidance for the current
decision: taking the action "{action}" would be better.
```

The field `action` is the preferred action a_z^+ in the positive branch and the inferior action a_z^- in the negative branch. Both branches use exactly the same guidance wording; the prompt does not expose the branch label. ALFWorld collapses whitespace in the action string; WebShop canonicalizes its action syntax and quotes it as a JSON string. CPAD+ and CPAD− use the corresponding single guidance branch. These augmented prompts are used only for training-time re-scoring. Rollouts and evaluation use the ordinary interaction prompts.

H Experimental Settings

Table 7 summarizes the common configuration of the available runs in Table 1 across two benchmarks and three Qwen2.5-Instruct scales. Table 8 lists the setting-dependent parameters.

Evaluation protocol. Evaluation uses 140 ALFWorld `val_seen` tasks, 134 `val_unseen` tasks, and 500 fixed WebShop goals, with one sampled trajectory per task and the training interaction limits. ALFWorld reports family and aggregate success; WebShop reports official partial-credit Score and exact Success. Main results use update 150; VANILLA uses GRPO at update 0. Except

Table 7: Common training and evaluation hyperparameters across the six benchmark–model settings. Clipping parameters apply to methods that use the clipped policy objective.

Parameter	Value
Training tasks per update / rollouts per task	16 / 8
Maximum training updates	150
Optimization epochs per rollout batch	1
Optimizer	AdamW
Learning rate / schedule	10^{-6}; constant, no warmup
Weight decay	0.01
Gradient-norm clipping	1.0
Policy-ratio clipping (lower / upper)	0.2 / 0.2
Dual-clipping coefficient	3.0
Reference KL coefficient / estimator	0.01 / low-variance KL
Entropy coefficient	0.001
Invalid-action penalty coefficient	0.1
Interaction history	Up to 2 observation–action pairs
Maximum response length	512 tokens per decision
Training / evaluation temperature	1.0 / 0.4
Sampling / top-p / top-k (train and eval.)	Enabled / 1.0 / disabled
Data / environment / rollout seeds	0 / 0 / 0
Evaluation schedule	Update 0, then every 5 updates
Checkpoint interval	10 updates
Rollout precision	bfloat16

Table 8: Settings for the two benchmarks and three model scales. Mini-batches count decision-response rows before data-parallel partitioning; micro-batches count rows per device. Slash-separated values indicate method-dependent choices within a setting.

	ALFWorld			WebShop		
Parameter	1.5B	3B	7B	1.5B	3B	7B
Maximum interaction steps	50	50	50	15	15	15
Ordinary prompt limit (tokens)	2048	2048	2048	4096	4096	4096
Privileged prompt limit (CPAD)	3072	3072	3072	4096	4096	4096
Actor mini-batch size	256	256	256	64	64	64
Actor micro-batch size	32	16/32	4/8	8/16	4/8	4/8
Rollout log-probability micro-batch	32	16/32	4/8	16/32	8/16	4/8/16
Rollout tensor parallelism	1	2	4	1	2	4

for VANILLA, Table 1 reports means over five evaluations of the same checkpoint, with sample standard deviations additionally shown for ALFWorld seen/unseen success rates and WebShop score and success rate. Figures 1(b) and 4(a) use the corresponding means, while learning curves and appendix sensitivity analyses retain the measurements collected during training.

Baseline implementations and settings. Under the shared protocol above, RLOO, GRPO, HGPO, GiGPO, and GraphGPO use `verl-agent` (Feng et al., 2025); SDAR, AgentOPSD, and Agent-G^2 use their respective public implementations. SDAR and AgentOPSD use the supplied skill libraries, and Agent-G^2 uses the released expert trajectories. HGPO and GiGPO use discount factor 0.95, with group advantages standardized on ALFWorld and mean-centered on WebShop. GraphGPO uses discount factors 0.1 and 0.2, respectively. SDAR uses distillation weight 0.01 and gate coefficient 5. AgentOPSD uses signed reweighting with mixing weight 0.5 and multiplier band 0.2, at turn level on ALFWorld and token level on WebShop. Agent-G^2 uses Gaussian prefix sampling capped at 0.8, sampling seed 1, and a prefix cross-entropy weight of 0.5.

CPAD settings. CPAD uses $\lambda = 1$, contrast bound $c_\Delta = 5$, and unit weights for both guidance branches; the one-sided variants set the unused weight to zero. CPAD and the matched GRPO baseline use success-reward scale $c_R = 10$, invalid-action penalty $c_{\text{inv}} = 0.1$, and normalization constant $\epsilon = 10^{-6}$ on both benchmarks (Appendix A). The privileged prompt limits in Table 8 apply to all

CPAD variants. Appendix B varies λ on WebShop 1.5B. Default matching uses $h = 0$, two distinct trajectories per decision group, and one supporting trajectory per retained action. Appendix D varies these support thresholds, matching history, and c_Δ, while keeping $\lambda = 1$. Appendix E changes the correction granularity while retaining the default selector and contrast settings; the token-shuffle control uses shuffle seed 0.

Execution settings. Training uses four GPUs per run, except for Agent-G^2 at all three ALFWorld scales and ALFWorld 7B SDAR, which use eight. Actor optimizer offloading is disabled at 1.5B and 3B, enabled on ALFWorld 7B, and enabled on WebShop 7B except for GRPO.

I Theoretical Properties of Paired Privileged Signals

We analyze shared-context cancellation and the policy direction induced by the contrast in Equation 9. The cancellation analysis builds on contrastive self-distillation (Pan et al., 2026; Heakl et al., 2026).

I.1 Notation

Fix an occurrence (i, t) and a valid response position $k \in \mathcal{K}_{i,t}$. Throughout this section, the full ordinary context $h^0_{i,t}$, prefix $y_{i,t,<k}$, selected action pair, and frozen parameters $\bar{\theta}$ are held fixed. Define the token distributions

$$q^b(v) = \pi_{\bar{\theta}}\left(v \mid h^b_{i,t}, y_{i,t,<k}\right), \qquad b \in \{0, +, -\}, \quad v \in \mathcal{V},$$

where $\mathcal{V}$ is the finite token vocabulary and all three distributions have positive support. Extend the sampled-token components to candidate tokens by

$$d^+(v) = \log \frac{q^+(v)}{q^0(v)}, \qquad d^-(v) = \log \frac{q^0(v)}{q^-(v)}, \qquad \Delta(v) = d^+(v) + d^-(v). \tag{15}$$

At $v = y_{i,t,k}$, these equal $d^+_{i,t,k}$, $d^-_{i,t,k}$, and $\Delta_{i,t,k}$, respectively. Their expectations under q^0 can be expressed using the Kullback–Leibler divergence D_{KL}:

$$\begin{aligned} \mathbb{E}_{q^0} d^+ &= -D_{\mathrm{KL}}(q^0 \| q^+) \le 0, \qquad \mathbb{E}_{q^0} d^- = D_{\mathrm{KL}}(q^0 \| q^-) \ge 0, \\ \mathbb{E}_{q^0} \Delta &= D_{\mathrm{KL}}(q^0 \| q^-) - D_{\mathrm{KL}}(q^0 \| q^+). \end{aligned}$$

The paired signal thus measures relative compatibility with the preferred- and inferior-action contexts.

For any candidate-token function f, define its centered version and weighted squared norm as

$$\mathcal{C} f = f - \mathbb{E}_{q^0} f, \qquad \|f\|^2_{q^0} = \mathbb{E}_{v \sim q^0}[f(v)^2].$$

For this analysis, centering removes token-independent constants when comparing directions. Indeed, for $q_\theta(v) = \pi_\theta(v \mid h^0_{i,t}, y_{i,t,<k})$, $\mathbb{E}_{q^0}[\nabla_\theta \log q_\theta(v)]_{\theta=\bar{\theta}} = 0$. Thus centering a fixed raw signal leaves its local expected score-function gradient unchanged.

I.2 Cancellation Compared with Either One-Sided Signal

Assumption I.1 (Shared context perturbation)**.** There are a preference-specific function $u : \mathcal{V} \to \mathbb{R}$ and a shared context perturbation $\nu : \mathcal{V} \to \mathbb{R}$ such that

$$q^\pm(v) = \frac{q^0(v) \exp\{\nu(v) \pm u(v)\}}{Z_\pm}, \qquad Z_\pm = \sum_{w \in \mathcal{V}} q^0(w) \exp\{\nu(w) \pm u(w)\}. \tag{16}$$

The target evidence u is held fixed as the shared perturbation ν varies.

The assumption separates the response to having privileged guidance, represented by ν, from equal-strength responses to its preferred and inferior content, represented by u and $-u$. Identical prompt wording motivates the shared component; the following proposition isolates its contribution to the signal.

Proposition I.1 (Strict reduction of shared-context error)**.** *Under Assumption I.1,*

$$\mathcal{C}d^+ = \mathcal{C}u + \mathcal{C}\nu, \qquad \mathcal{C}d^- = \mathcal{C}u - \mathcal{C}\nu, \qquad \tfrac{1}{2}\mathcal{C}\Delta = \mathcal{C}u. \tag{17}$$

Consequently, at the same scale of preference evidence,

$$\left\|\tfrac{1}{2}\mathcal{C}\Delta - \mathcal{C}u\right\|_{q^0}^2 = 0, \qquad \left\|\mathcal{C}d^{\pm} - \mathcal{C}u\right\|_{q^0}^2 = \|\mathcal{C}\nu\|_{q^0}^2. \tag{18}$$

The paired signal has strictly smaller error than either one-sided signal whenever ν is nonconstant on the support of q^0.

Proof. Taking logarithms in Equation 16 gives

$$d^+(v) = \nu(v) + u(v) - \log Z_+, \qquad d^-(v) = -\nu(v) + u(v) + \log Z_-.$$

The constants $\log Z_\pm$ vanish under $\mathcal{C}$, giving Equation 17. Subtracting $\mathcal{C}u$ and taking squared norms yields Equation 18. Positivity is strict exactly when $\|\mathcal{C}\nu\|_{q^0}^2 > 0$. □

The factor $1/2$ matches the preference coefficient for this comparison of raw signals; canonical CPAD still uses the sum Δ in Equation 9. More generally,

$$\mathcal{C}(w_+ d^+ + w_- d^-) = (w_+ + w_-)\mathcal{C}u + (w_+ - w_-)\mathcal{C}\nu.$$

Among nonzero linear combinations with fixed nonnegative weights, equal weights are necessary and sufficient to cancel every shared perturbation. If the two log densities additionally contain branch-specific errors e^+ and e^-, the residual becomes $\frac{1}{2}\mathcal{C}(e^+ - e^-)$; its norm is at most $\frac{1}{2}(\|\mathcal{C}e^+\|_{q^0} + \|\mathcal{C}e^-\|_{q^0})$. Thus the exact advantage concerns shared perturbations, with remaining error governed by disagreement between the branches.

I.3 When the Contrast Gives a Useful Policy Direction

Define the continuation success value of a candidate token, with all subsequent generation and interaction following the frozen ordinary policy, as

$$Q(v) = \Pr_{\pi_{\bar{\theta}}}\left(R_i = 1 \mid h_{i,t}^0, y_{i,t,<k}, y_{i,t,k} = v\right), \qquad V = \mathbb{E}_{q^0} Q. \tag{19}$$

These quantities define the value criterion for the following analysis. For $g = g_{i,t}$ and $c_\Delta > 0$, extend Equation 10 to $\widetilde{\Delta}(v) = g\ \mathrm{sg}[\mathrm{clip}(\Delta(v), -c_\Delta, c_\Delta)]$. All teacher scores and g remain fixed during the following update.

Proposition I.2 (Value improvement under an aligned contrast)**.** *Suppose the raw contrast preserves the ordering of continuation values:*

$$(Q(v) - Q(w))(\Delta(v) - \Delta(w)) \geq 0 \qquad \textit{for all } v, w \in \mathcal{V}. \tag{20}$$

For $\eta > 0$ and the nonnegative correction strength λ, consider the idealized KL-regularized update

$$q_\eta = \arg\max_q \left\{\lambda \mathbb{E}_q \widetilde{\Delta} - \frac{1}{\eta} D_{\mathrm{KL}}(q \| q^0)\right\}. \tag{21}$$

Then $\mathbb{E}_{q_\eta} Q \geq V$. For $\lambda > 0$, improvement is strict if there are candidates with different values and strictly ordered corrected scores.

Proof. The KL-regularized objective has the standard exponential-tilting solution (Rafailov et al., 2023),

$$q_\eta(v) = \frac{q^0(v) \exp\{\eta\lambda\widetilde{\Delta}(v)\}}{\sum_w q^0(w) \exp\{\eta\lambda\widetilde{\Delta}(w)\}}. \tag{22}$$

To verify optimality, substitute this expression into Equation 21: its objective is a constant minus $\eta^{-1} D_{\mathrm{KL}}(q \| q_\eta)$, uniquely maximized at q_η. Differentiating its expected value while keeping Q and the scores fixed,

$$\begin{aligned}\frac{d}{d\eta}\mathbb{E}_{q_\eta} Q &= \lambda\, \mathrm{Cov}_{q_\eta}(Q, \widetilde{\Delta}) \\ &= \frac{\lambda}{2}\mathbb{E}_{v,w \overset{\text{iid}}{\sim} q_\eta}\left[(Q(v) - Q(w))(\widetilde{\Delta}(v) - \widetilde{\Delta}(w))\right] \geq 0.\end{aligned} \tag{23}$$

Clipping is nondecreasing, and the gate is constant within this prefix, so Equation 20 makes each product nonnegative. Integrating from $\eta = 0$, where the continuous extension is $q_0 = q^0$, proves the claim. Positive support makes the inequality strict under the stated condition. □

Assumption I.1 satisfies Equation 20 whenever $u(v) = \beta Q(v)$ for some $\beta > 0$: $\Delta(v) = 2\beta Q(v) + \log Z_- - \log Z_+$. Thus a common perturbation can be arbitrarily large without altering the value ordering of the raw paired signal. Each one-sided signal retains the token-dependent term $+\nu$ or $-\nu$. Adding the true token advantage $Q(v) - V$ to the linear objective in Equation 21 defines a distinct augmented update,

$$q_\eta^{\mathrm{aug}}(v) \propto q^0(v) \exp\{\eta[Q(v) - V + \lambda\widetilde{\Delta}(v)]\}.$$

Its value derivative is $\mathrm{Var}_{q_\eta^{\mathrm{aug}}}(Q) + \lambda \,\mathrm{Cov}_{q_\eta^{\mathrm{aug}}}(Q, \widetilde{\Delta}) \geq 0$, so it also preserves improvement.

A sufficient probabilistic interpretation. Posterior-compatible teachers provide another route to value alignment, related to Bayesian interpretations of self-distillation (Shen et al., 2026; Heakl et al., 2026). If, at this fixed prefix, q^+ and q^- equal the ordinary policy's token distributions conditioned on $R_i = 1$ and $R_i = 0$, respectively, and $0 < Q(v), V < 1$, Bayes' rule gives

$$q^+(v) = q^0(v)\frac{Q(v)}{V}, \qquad q^-(v) = q^0(v)\frac{1 - Q(v)}{1 - V}.$$

Hence

$$\Delta(v) = \log \frac{Q(v)}{1 - Q(v)} - \log \frac{V}{1 - V}. \tag{24}$$

The log-odds function is strictly increasing, so the contrast has the same sign as $Q(v) - V$ and preserves value ordering. This posterior model provides a sufficient construction of aligned teachers. CPAD constructs its teachers from preferred- and inferior-action guidance.

Approximate alignment. For an untruncated signal of the form $\Delta(v) = \alpha Q(v) + e(v) + c_0$, where $\alpha > 0$ and c_0 is token-independent, define the untruncated, ungated update $q_\eta^{\mathrm{raw}}(v) \propto q^0(v) e^{\eta\lambda\Delta(v)}$. The superscript distinguishes it from the gated, clipped q_η in Equation 22. Its initial value derivative obeys

$$\frac{d}{d\eta}\mathbb{E}_{q_\eta^{\mathrm{raw}}} Q\bigg|_{\eta=0} \geq \lambda\sigma_Q(\alpha\sigma_Q - \sigma_e), \qquad \sigma_f^2 = \mathrm{Var}_{q^0}(f). \tag{25}$$

This follows from $\mathrm{Cov}(Q, \Delta) = \alpha \,\mathrm{Var}(Q) + \mathrm{Cov}(Q, e)$ and Cauchy–Schwarz. A sufficiently small update is therefore useful when $\lambda > 0$, $\sigma_Q > 0$, and $\sigma_e < \alpha\sigma_Q$. This relates the usefulness of a selected decision to the amount of value variation relative to signal error.

I.4 Implications for CPAD

Proposition I.1 shows how equal weighting removes shared context perturbations from the centered raw preference signal. Proposition I.2 connects aligned contrast to continuation-value improvement. To examine the local effect of pairing, consider untruncated, ungated updates with unit correction strength, $u = \beta Q$, and equal preference scale. The initial value derivative for $\Delta/2$ is $\beta \,\mathrm{Var}_{q^0}(Q)$, while those for d^+ and d^- are $\beta \,\mathrm{Var}_{q^0}(Q) + \mathrm{Cov}_{q^0}(Q, \nu)$ and $\beta \,\mathrm{Var}_{q^0}(Q) - \mathrm{Cov}_{q^0}(Q, \nu)$, respectively. The paired value derivative is their mean and is independent of the shared perturbation.

Clipping can alter centered-score cancellation through saturation induced by normalization constants. CPAD optimizes the sampled-token correction with a shared-parameter clipped actor objective.