DL-OPSD: DUAL-LEVEL SELECTIVE ON-POLICY SELF-DISTILLATION FOR SKILL INTERNALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation combined with reinforcement learning offers a route to internalizing external skills in language agents, removing the need for skill retrieval at inference. However, local teacher–student prediction gaps alone are insufficient to determine which behaviors deserve reinforcement through distillation. We find that, when teacher–student gaps are comparable, original actions yield positive average gains in continuation success over student-sampled alternatives on positive-advantage trajectories, but negative gains on negative-advantage trajectories. This observation suggests that trajectory outcomes inform supervision selection beyond local prediction gaps, motivating **D**ual-**L**evel Selective **O**n-**P**olicy **S**elf-**D**istillation (DL-OPSD). At the trajectory level, DL-OPSD applies auxiliary distillation only to trajectories with positive group-relative advantage, while retaining reinforcement learning updates across all trajectories. At the token level, it selects positions where the teacher assigns a higher probability to the sampled token than the student does, and weights the supervision according to student uncertainty and teacher certainty. Skills are provided only to the teacher during training. Across ALFWorld, Search-QA, and WebShop, DL-OPSD outperforms GRPO and SDAR on the primary task metrics, improving Qwen3-1.7B's ALFWorld success rate from 60.9% under GRPO to 70.3%. Controlled experiments show that trajectory selection improves gradient alignment between distillation and reinforcement learning, while token-level weighting further improves task performance, supporting the complementary roles of the two levels.

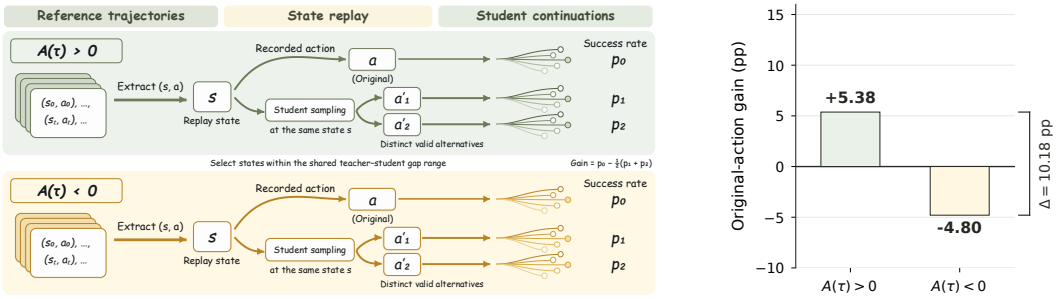
1 INTRODUCTION

Skills equip large language model (LLM) agents with reusable guidance for planning, search, and tool use, helping them tackle complex, multi-step tasks effectively (Xia et al., 2026; Lu et al., 2026b). However, repeatedly retrieving and incorporating skills at inference time incurs additional context overhead and latency, making agent performance sensitive to skill availability and retrieval quality. Skill internalization aims to transfer this external guidance into the model's own policy, enabling agents to perform tasks without access to skills (Lu et al., 2026b). On-policy self-distillation (OPSD) with skills as privileged information offers a natural route toward this goal (Zhao et al., 2026; Lu et al., 2026a). The same model acts as both student and teacher: the student receives no skills, while the teacher receives additional skills and provides guidance at student-visited states. Combined with reinforcement learning (RL), this approach provides complementary learning signals from both fine-grained self-guidance and the outcomes of its own interactions.

However, privileged teacher guidance can conflict with outcome-based RL, such as GRPO (Shao et al., 2024). On trajectories with negative group-relative advantage, RL discourages sampled behaviors, while auxiliary distillation may reinforce tokens favored by the skill-conditioned teacher. Although the teacher receives additional skill information, its predictions are not conditioned on the trajectory outcome. To examine this phenomenon, we compare the downstream success of actions with comparable teacher–student gaps. Relative to student-sampled alternatives at the same state, the original actions from positive-advantage trajectories yield an average continuation success gain of $+5.4\%$, whereas those from negative-advantage trajectories yield -4.8% (Figure 1).

This suggests that local teacher–student signals alone may be insufficient to determine where privileged supervision should be applied. Prior work uses teacher–student prediction differences to

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) Computing original-action gain

(b) Gain by trajectory advantage

Figure 1: **Original-action gain at comparable teacher–student gaps.** (a) Original-action gain compares continuation success after the original action with that after student-sampled alternatives from the same state. (b) Average gain for positive- and negative-advantage trajectories within the common teacher–student gap interval. Error bars indicate 95% confidence intervals.

modulate policy updates or auxiliary distillation (Yang et al., 2026; Lu et al., 2026a), and explores token importance and outcome-based sample routing (Xu et al., 2026; Li et al., 2026a). Yet our observation reveals a challenge when auxiliary distillation is combined with RL: similar local prediction gaps can correspond to substantially different downstream action gains depending on trajectory outcomes. This raises a central question:

When self-guidance and execution feedback disagree, how should a model learn from itself?

Motivated by this, we introduce ***D**ual-**L**evel Selective **O**n-**P**olicy **S**elf-**D**istillation (DL-OPSD)*, which selectively applies auxiliary distillation at both the trajectory and token levels. At the trajectory level, distillation is restricted to positive-advantage trajectories, while RL continues to learn from all trajectories. At the token level, we retain positions with teacher support, where the teacher assigns a higher probability to the sampled token than the student does, and weight their supervision according to student uncertainty and teacher certainty. Skills are provided only to the teacher during training, allowing the student to perform tasks without skill access at inference time.

We evaluate DL-OPSD on ALFWorld, Search-QA, and WebShop across multiple model sizes, demonstrating improvements in student task performance without skill access. Through matched-budget random trajectory selection, entropy-factor ablations, and entropy-factor permutation, we further examine how trajectory outcomes and local predictive signals contribute to effective distillation. Our results suggest that trajectory outcomes should guide the allocation of self-distillation supervision. Our main contributions are as follows:

- We identify a limitation of relying on local self-guidance alone: comparable teacher–student prediction gaps can correspond to opposite downstream action gains depending on trajectory outcomes, showing that local teacher signals alone do not determine whether sampled behavior is worth reinforcing.
- To address this limitation, we propose DL-OPSD, which restricts auxiliary distillation to positive-advantage trajectories while retaining RL updates on all trajectories. Within selected trajectories, teacher support selects token positions, while student uncertainty and teacher certainty determine supervision weights.
- Experiments across multiple model sizes on ALFWorld, Search-QA, and WebShop demonstrate that DL-OPSD improves task performance without skill access. Ablation studies further examine the contributions of trajectory-level selection and token-level weighting.

2 PRELIMINARIES

We first formalize the interaction setting and teacher-student setup for OPSD. Given a task input x, the policy π_θ generates a group of G interaction trajectories without access to skills. Let $y_i = (y_{i,1}, \ldots, y_{i,T_i})$ denote the policy-generated token sequence of trajectory i, and let R_i denote its trajectory-level return from the environment. We compute the training losses over policy-generated

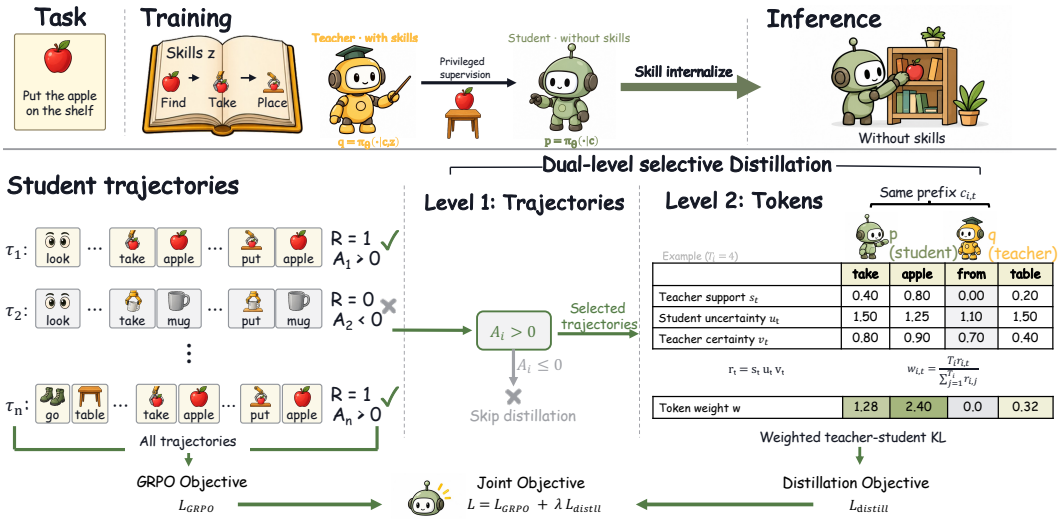


Figure 2: **Overview of DL-OPSD.** The student generates on-policy trajectories without skills, while the teacher receives additional skills and evaluates the same prefixes. Group-relative advantages select trajectories for auxiliary distillation. Within selected trajectories, teacher support selects token positions, and student uncertainty and teacher certainty refine supervision weights. GRPO uses all trajectories, and the resulting student operates without skills at inference.

tokens, while environment observations and tool responses serve as interaction context. For each token $y_{i,t}$, let $c_{i,t}$ denote the interaction context preceding its generation, including the task input, previous policy-generated tokens, and environment observations. The student and teacher share the same policy parameters but differ in their available context: the student conditions only on $c_{i,t}$, whereas the teacher additionally receives a retrieved skill z:

$$p_{i,t} = \pi_\theta(\cdot \mid c_{i,t}), \qquad q_{i,t} = \pi_\theta(\cdot \mid c_{i,t}, z). \tag{1}$$

Thus, both predict the next token from the same student-generated prefix, with privileged skill information available only to the teacher. During distillation, the teacher distribution $q_{i,t}$ is detached and serves as a fixed supervision target.

3 Method

DL-OPSD allocates auxiliary distillation using complementary signals at two levels. Trajectory outcomes determine the distillation scope, while local teacher–student predictions determine supervision positions and weights within that scope. Meanwhile, RL (instantiated by GRPO) continues to learn from all trajectories (Figure 2).

3.1 Trajectory-Level Selection

The continuation analysis in Figure 1 reveals that similar teacher–student signals can correspond to markedly different action quality depending on trajectory outcomes. This suggests that local teacher preference alone is insufficient to determine where privileged supervision should be applied, and that execution outcomes provide a complementary signal for identifying more favorable contexts for distillation. Motivated by this observation, we restrict auxiliary distillation to trajectories whose returns exceed the group average.

For the G trajectories generated for the same task, we compute the group-relative advantage as:

$$A_i = \frac{R_i - \overline{R}}{\sigma_R + \epsilon}, \qquad \overline{R} = \frac{1}{G}\sum_{j=1}^{G} R_j, \tag{2}$$

where R_i denotes the return of trajectory i, σ_R is the standard deviation of group returns, and ϵ ensures numerical stability. We then instantiate trajectory selection with the binary gate

$$g_i = \mathbf{1}[A_i > 0]. \tag{3}$$

Thus, privileged distillation is applied only to positive-advantage trajectories, while trajectories with $A_i \leq 0$ remain in the GRPO objective but receive no auxiliary distillation.

3.2 Token-Level Selection and Weighting

Trajectory selection incorporates evidence from execution outcomes, but trajectory-level feedback is too coarse to determine how to allocate supervision across individual tokens, as better trajectory performance does not imply that each token warrants equal supervision. Within selected trajectories, we therefore identify positions where the privileged teacher provides positive support and allocate supervision according to student uncertainty and teacher certainty.

Teacher support. Within selected trajectories, we seek behaviors that the student already generates and that are further supported by the skill-conditioned teacher, making them natural targets for internalizing privileged guidance into the student policy. To identify such behaviors, we compare the probabilities assigned by the teacher and student to the sampled token under the same student-generated prefix:

$$\Delta_{i,t} = \log q_{i,t}(y_{i,t}) - \log p_{i,t}(y_{i,t}). \tag{4}$$

A positive gap indicates that conditioning on skill context increases the model's support for the sampled token. We therefore retain only positions with positive teacher support for distillation, using a hard gate to assign zero weight otherwise:

$$s_{i,t} = \mathbf{1}[\Delta_{i,t} > \delta]\, \phi(\Delta_{i,t}). \tag{5}$$

Here, the threshold $\delta \geq 0$ filters near-zero gaps, and the monotonic function $\phi(\cdot)$ maps retained gaps to $(0, 1]$, giving stronger teacher support a larger weight. More details about scaling and clipping can be found in Appendix B.1.

Student uncertainty and teacher certainty. Positive teacher support identifies positions where privileged information favors the student's sampled behavior, but it does not indicate how strongly each retained position should be supervised. We therefore further account for the student's need for guidance and the teacher's confidence in that guidance. We assign greater weight to positions where the student is more uncertain and the teacher is more certain. We capture these complementary signals using normalized student and teacher entropy:

$$u_{i,t} = 1 + \eta\, \widetilde{H}_S(p_{i,t}), \qquad v_{i,t} = 1 - \widetilde{H}_T(q_{i,t}), \tag{6}$$

where $\widetilde{H}_S$ and $\widetilde{H}_T$ denote normalized student and teacher entropy, with normalization details in Appendix B.2. The coefficient $\eta \geq 0$ controls the contribution of student uncertainty. Consequently, supervision is emphasized at positions where the student is more uncertain and the teacher is more certain. Section 5.2 further examines the effect of this allocation on student learning.

We combine the three factors multiplicatively and normalize the resulting weights within each selected trajectory:

$$r_{i,t} = s_{i,t} u_{i,t} v_{i,t}, \qquad w_{i,t} = \frac{T_i r_{i,t}}{\sum_{j=1}^{T_i} r_{i,j}}. \tag{7}$$

Equation (7) separates token-level selection from allocation: positive teacher support determines whether a position receives distillation, while student uncertainty and teacher certainty determine its relative supervision weight. When the total local weight is nonzero, normalization ensures $\sum_t w_{i,t} = T_i$, preserving an average weight of one over all valid policy-generated positions in the trajectory. Otherwise, we set all $w_{i,t}$ for that trajectory to zero.

3.3 Training Objective

The two selection levels determine where and how strongly auxiliary distillation is applied. At each selected position, we use the forward KL divergence to bring the student's next-token distribution closer to the teacher's:

$$d_{i,t} = D_{\mathrm{KL}}(q_{i,t} \| p_{i,t}). \tag{8}$$

Combining trajectory selection and token weighting gives the auxiliary distillation loss:

$$\mathcal{L}_{\text{distill}} = \frac{\sum_i \sum_{t=1}^{T_i} g_i w_{i,t} d_{i,t}}{\sum_i T_i}. \tag{9}$$

The sums range over all trajectories in the training batch, and the denominator counts all valid policy-generated tokens. Within-trajectory normalization controls the relative allocation of supervision across tokens. The batch-level denominator does not change with trajectory selection, so the supervision weight removed from excluded trajectories is not redistributed to the remaining trajectories. If no trajectory in the batch has positive advantage, the distillation loss is zero.

The full training objective is

$$\mathcal{L} = \mathcal{L}_{\text{GRPO}} + \lambda \mathcal{L}_{\text{distill}}, \tag{10}$$

where λ controls the contribution of auxiliary distillation. GRPO updates the policy using execution feedback from all trajectories, while auxiliary distillation transfers selected skill guidance into the student. Skills are provided only to the teacher during training, and the student requires no skill access at inference. Algorithm 1 summarizes the complete training procedure.

Algorithm 1: DL-OPSD training.

Require: policy π_θ, task set $\mathcal{S}$, skill library $\mathcal{E}$, group size G, coefficient λ.

for each training iteration do
  Sample a task batch $\mathcal{B} \subset \mathcal{S}$
  for each $x \in \mathcal{B}$ **do**
    Retrieve skill $z \in \mathcal{E}$; sample $\{y_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x)$ without skills
    Obtain returns R_i; $A_i \leftarrow (R_i - \overline{R})/(\sigma_R + \epsilon)$; $g_i \leftarrow \mathbf{1}[A_i > 0]$
    for each valid position (i, t) **do**
      $p_{i,t} \leftarrow \pi_\theta(\cdot \mid c_{i,t})$; $q_{i,t} \leftarrow \text{sg}[\pi_\theta(\cdot \mid c_{i,t}, z)]$
      $\Delta_{i,t} \leftarrow \log q_{i,t}(y_{i,t}) - \log p_{i,t}(y_{i,t})$
      $s_{i,t} \leftarrow \mathbf{1}[\Delta_{i,t} > \delta]\phi(\Delta_{i,t})$
      $u_{i,t} \leftarrow 1 + \eta \widetilde{H}_S(p_{i,t})$; $v_{i,t} \leftarrow 1 - \widetilde{H}_T(q_{i,t})$
      $r_{i,t} \leftarrow s_{i,t} u_{i,t} v_{i,t}$; $d_{i,t} \leftarrow D_{\text{KL}}(q_{i,t} \| p_{i,t})$
    end for; $w_{i,t} \leftarrow \text{sg}[T_i r_{i,t} / \sum_j r_{i,j}]$, or 0 if $\sum_j r_{i,j} = 0$
  end for
  $\mathcal{L}_{\text{distill}} \leftarrow \sum_{i,t} g_i w_{i,t} d_{i,t} / \sum_i T_i$
  Compute $\mathcal{L}_{\text{GRPO}}$ from all trajectories in $\mathcal{B}$
  $\theta \leftarrow \text{Update}(\theta, \mathcal{L}_{\text{GRPO}} + \lambda \mathcal{L}_{\text{distill}})$
end for

Output: policy π_θ.

4 Experiments

4.1 Experimental Setup

We evaluate DL-OPSD on ALFWorld (Shridhar et al., 2020), Search-QA (Jin et al., 2025), and WebShop (Yao et al., 2022) with Qwen3-1.7B (Yang et al., 2025), Qwen2.5-3B-Instruct, and Qwen2.5-7B-Instruct (Yang et al., 2024). We compare training-free, post-training, and hybrid reinforcement-learning and distillation baselines. Each trainable method samples eight trajectories per task and runs for 300 policy updates. We report the best validation checkpoint. See Appendix C for details.

4.2 Main Results

Overall Performance. Table 1 summarizes task performance across three model sizes and three environments. Notably, DL-OPSD outperforms GRPO and SDAR on the primary task metrics across all settings. For example, our method improves ALFWorld success from 60.9% to 70.3% with Qwen3-1.7B under GRPO. These results show that selective skill distillation provides additional benefits beyond learning from environment returns. In particular, the improvements over SDAR indicate that token-level predictive information provides an additional signal for allocating auxiliary supervision beyond trajectory outcomes alone.

Table 1: **Performance across ALFWorld, Search-QA, and WebShop.** Values are rounded to one decimal. Within each backbone block, the best result in each column is shown in bold and the second-best is underlined. Columns corresponding to the primary evaluation metric of each benchmark are shaded. A superscript * denotes evaluation with skills.

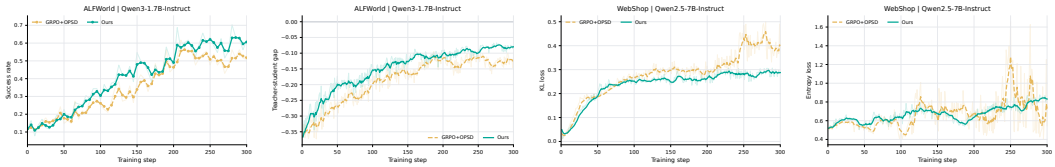
	ALFWorld							Search-QA								WebShop	
Method	**Pick**	**Look**	**Clean**	**Heat**	**Cool**	**Pick2**	**Avg.**	**NQ**	**Triv**	**Pop**	**Hotp**	**2Wk**	**MuS**	**Bam**	**Avg.**	**Score**	**Acc.**
Qwen2.5-3B-Instruct																	
Vanilla	51.7	16.7	16.1	0.0	17.4	10.0	21.1	26.2	46.2	32.8	26.4	24.0	8.2	18.4	26.9	3.3	0.8
Skill-Prompt*	37.9	50.0	32.3	5.3	0.0	5.0	20.3	28.8	43.6	34.4	23.2	22.4	7.0	22.4	26.4	4.9	0.8
Skill-GRPO	96.0	61.5	69.6	52.6	40.9	42.3	60.9	46.0	59.8	51.2	41.8	39.8	16.8	31.2	42.1	86.3	72.7
Skill-GRPO*	90.7	75.0	95.8	66.7	84.2	61.1	82.8	48.2	58.1	39.9	38.5	38.9	15.8	36.7	39.8	87.4	75.8
OPSD	51.7	33.3	25.8	5.3	21.7	15.0	26.6	26.2	46.2	32.8	26.4	24.0	8.2	18.4	26.9	5.7	2.3
GRPO	**100.0**	81.8	95.7	82.4	87.0	88.0	90.6	46.6	61.0	52.6	42.0	39.6	17.4	33.6	42.8	88.2	82.8
GRPO+OPSD	96.0	80.0	88.9	87.5	94.7	73.1	86.7	**48.6**	61.2	**53.0**	42.6	42.8	16.2	31.2	43.6	91.6	80.5
RLSD	96.6	83.3	90.3	89.5	82.6	80.0	88.3	46.4	60.2	52.0	40.8	42.0	19.4	36.8	43.2	90.8	81.3
SKILL0	97.3	84.6	88.9	93.3	79.2	**90.5**	89.8	45.4	55.5	44.5	43.9	42.2	16.4	**36.9**	41.1	88.3	78.9
SDAR	**100.0**	76.9	**96.3**	93.8	63.6	81.0	86.7	47.0	59.2	47.6	41.2	42.2	**21.4**	36.0	42.8	86.3	78.9
DL-OPSD (Ours)	92.6	**87.5**	92.3	**100.0**	**96.2**	75.0	**91.4**	45.8	**62.0**	52.0	**44.8**	**43.6**	18.8	36.2	**44.2**	**93.1**	**85.9**
Qwen2.5-7B-Instruct																	
Vanilla	34.5	33.3	19.4	10.5	0.0	5.0	16.4	26.8	50.6	31.8	28.2	28.2	8.8	27.2	29.0	3.3	0.8
Skill-Prompt*	51.7	50.0	35.5	21.1	4.3	0.0	26.6	29.4	49.0	37.4	30.6	29.4	13.4	31.2	31.5	1.7	0.8
Skill-GRPO	92.9	75.0	62.5	78.6	76.0	40.0	70.3	49.6	63.6	**56.6**	44.8	46.8	20.0	44.0	46.8	89.1	81.3
Skill-GRPO*	**100.0**	83.3	**100.0**	91.7	78.9	77.8	91.4	50.6	61.2	51.7	**50.4**	45.2	23.8	42.7	47.0	90.2	82.8
OPSD	58.6	66.7	25.8	26.3	21.7	15.0	32.8	26.8	50.6	31.8	28.2	28.2	8.8	27.2	29.0	3.3	0.8
GRPO	96.7	85.7	**100.0**	84.6	81.0	75.0	88.3	50.2	62.0	52.6	46.2	44.6	22.4	44.8	46.3	87.9	82.0
GRPO+OPSD	96.8	80.0	92.9	90.9	82.8	78.9	88.3	**50.8**	63.4	52.0	46.8	45.8	23.8	44.0	47.0	87.5	82.8
RLSD	97.5	**100.0**	96.3	**100.0**	62.5	84.6	88.3	49.0	62.6	53.4	45.6	**50.6**	23.6	46.4	47.4	89.0	83.6
SKILL0	**100.0**	71.4	92.3	94.4	**88.9**	**90.0**	92.2	46.4	**65.0**	43.2	48.1	43.7	22.3	45.3	44.8	89.1	83.6
SDAR	96.6	84.6	92.6	87.5	81.8	76.2	87.5	48.6	62.4	53.2	46.8	44.4	22.0	40.0	46.0	88.5	82.8
DL-OPSD (Ours)	96.3	96.9	96.2	**100.0**	80.8	87.5	**93.0**	48.6	**65.0**	55.0	47.6	43.8	**24.2**	**49.6**	**47.5**	**91.2**	**86.7**
Qwen3-1.7B																	
Vanilla	20.7	50.0	9.7	5.3	4.3	5.0	11.7	29.0	34.2	34.4	27.6	24.3	6.6	17.7	25.7	47.5	4.7
Skill-Prompt*	17.2	50.0	29.0	5.3	4.3	5.0	15.6	28.4	46.8	37.0	22.8	18.4	7.2	15.2	26.3	24.3	2.3
Skill-GRPO	55.2	50.0	32.3	36.8	21.7	35.0	37.5	42.0	57.0	50.8	35.4	34.8	14.2	30.4	38.6	80.1	54.7
Skill-GRPO*	58.6	66.7	51.6	36.8	26.1	30.0	43.8	45.4	53.5	47.3	40.8	39.3	**16.9**	**37.8**	40.4	84.3	65.6
OPSD	22.5	50.0	0.0	0.0	5.0	10.5	15.6	6.4	12.8	11.6	6.6	11.2	1.6	7.2	8.3	50.2	10.2
GRPO	69.7	58.3	60.7	75.0	62.5	36.8	60.9	43.6	53.8	49.6	39.2	**42.3**	16.1	36.4	40.6	**90.1**	79.7
GRPO+OPSD	86.2	**69.2**	59.3	75.0	45.5	33.3	61.7	42.8	53.6	40.6	39.8	41.7	15.8	36.2	38.9	83.6	72.7
RLSD	80.0	50.0	63.0	33.3	46.2	42.9	57.8	44.0	**58.2**	53.8	40.8	36.6	15.8	35.2	41.3	87.5	78.9
SKILL0	**93.3**	50.0	**69.2**	36.4	79.3	40.0	68.0	43.0	53.2	48.7	35.7	35.8	12.7	34.1	38.0	88.1	73.4
SDAR	79.4	33.3	60.0	**82.4**	65.0	38.1	64.1	38.5	54.7	49.4	37.2	36.9	13.7	29.3	38.0	80.6	70.3
DL-OPSD (Ours)	72.7	66.7	60.7	75.0	**87.5**	**57.9**	**70.3**	**46.8**	57.4	**54.0**	**41.0**	40.0	16.2	37.6	**42.4**	87.1	**81.3**

Task performance without skills. DL-OPSD outperforms Skill-GRPO, which uses skills during training but removes them at evaluation. On WebShop with Qwen2.5-3B, DL-OPSD achieves a success rate of 85.9% without skills at inference, outperforming both Skill-GRPO evaluated without skills (72.7%) and Skill-GRPO* evaluated with skills (75.8%).

Training dynamics. Figure 3 compares DL-OPSD with GRPO+OPSD during training. On ALFWorld with Qwen3-1.7B, DL-OPSD achieves higher success rates through much of the middle and later stages, while its mean teacher–student log-probability gap moves closer to zero. On WebShop with Qwen2.5-7B, GRPO+OPSD exhibits a pronounced late-stage increase in KL divergence from the reference policy, accompanied by large fluctuations in policy entropy. DL-OPSD maintains a flatter KL trajectory and smaller entropy fluctuations over the same period.

5 Analysis

In this section, we analyze how trajectory outcomes and local predictive information contribute to selective self-distillation. Specifically, we examine trajectory selection, entropy-based supervision allocation, and the complementary roles of these two levels of information.



(a) Success rate (b) Teacher–student gap (c) KL to reference (d) Policy entropy

Figure 3: **Training dynamics of DL-OPSD and GRPO+OPSD.** (a–b) ALFWorld with Qwen3-1.7B. (c–d) WebShop with Qwen2.5-7B. Faint curves show the raw logged values, and opaque curves show exponential moving averages.

5.1 Trajectory Outcomes and Distillation Selection

Trajectory outcomes complement the teacher–student gap. Building on the experiment in Section 1 (protocol in Appendix E.2), we quantify the group-level effect within the common teacher–student gap interval. Original-action gain averages $+5.4\%$ on positive-advantage trajectories and -4.8% on negative-advantage trajectories, yielding a group difference of 10.2% with a 95% cluster-bootstrap confidence interval of $[2.4, 18.3]$ (Figure 1). The direction holds throughout the common support region. Trajectory outcomes therefore carry behavioral information beyond the local teacher–student gap.

Trajectory outcomes inform distillation selection. Next, we examine whether using trajectory outcomes to select the distillation scope improves learning. All diagnostic variants start from the same step-150 GRPO checkpoint and continue training on ALFWorld for 50 updates. They apply sigmoid gap weighting at every token position and differ only in which trajectories are used for distillation: all trajectories, negative-advantage trajectories, positive-advantage trajectories, or a random-matched subset. The random-matched control uses the same number of trajectories and the same total normalized token weight as the positive-advantage set (Appendix E.3).

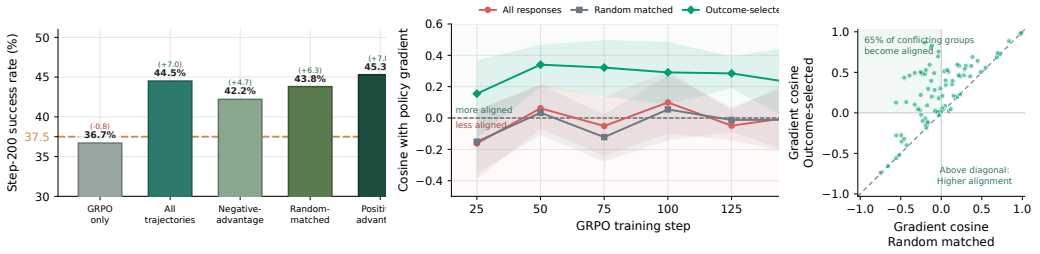
Positive-advantage selection achieves the highest diagnostic success rate, exceeding random-matched selection by 1.5 percentage points and all-trajectory distillation by 0.8 points (Figure 4(a)). The same comparison under the full training protocol confirms the benefit across all three environments: positive-advantage selection outperforms all-trajectory distillation by 0.8, 5.7, and 12.5 percentage points on ALFWorld, Search-QA, and WebShop, respectively (Table 2).

Table 2: **Trajectory selection with the full token-level rule.** Results follow the full training protocol. Columns report ALFWorld success rate, Search-QA exact match, and WebShop Acc., all in percentages. The shaded row denotes DL-OPSD.

Distillation scope	ALFWorld	Search-QA	WebShop
All trajectories	69.5	36.7	68.8
Negative-advantage trajectories	53.9	39.8	80.4
Positive-advantage trajectories	**70.3**	**42.4**	**81.3**

Trajectory selection improves gradient alignment. Another key question is how trajectory selection affects the interaction between auxiliary distillation and reinforcement learning. According six checkpoints from a single GRPO training run on ALFWorld, we keep the token-level rule fixed and compare all trajectories, random-matched selection, and outcome-selected trajectories on the same sampled trajectory groups. We measure the cosine similarity between the policy-loss gradients and distillation gradients for the first decision response of each trajectory. A positive cosine value indicates alignment, whereas a negative value indicates local optimization competition.

Outcome-selected trajectories achieve higher mean gradient alignment than both control conditions at every examined checkpoint examined(Figure 4). All trajectories and random-matched selection fluctuate around zero, whereas outcome-based selection more often yields positive alignment under the same supervision budget. A paired comparison further shows that 65% of trajectory groups with negative cosine under random-matched selection have positive cosine similarity under outcome-



(a) Trajectory selection (b) Gradient alignment across checkpoints (c) Budget-matched comparison

Figure 4: **Effect of trajectory selection on learning and gradient alignment.** (a) ALFWorld success rates after 50 additional updates. (b) Cosine similarity between policy-loss and distillation gradients across checkpoints. (c) Paired comparison of outcome-selected and random-matched supervision. Shaded bands show 95% bootstrap confidence intervals.

based selection. These findings provide an optimization-level explanation for the benefits of trajectory selection observed above.

5.2 Predictive Entropy for Supervision Allocation

Teacher support identifies tokens favored by the teacher, but does not capture the concentration of either predictive distribution. We therefore test whether student uncertainty and teacher certainty improve supervision at these positions. All configurations start from a shared step-150 checkpoint on ALFWorld and continue training until step 200. For each distillation variant, trajectory selection is fixed to positive-advantage trajectories ($A_i > 0$), and token positions are selected based on teacher support ($\Delta_{i,t} > \delta$). All variants use identical selection rules and trajectory-wise normalization, differing only in their local weighting factors.

We compare support-only weighting against three variants that additionally incorporate student uncertainty, teacher certainty, or both. The shuffled entropy jointly permutes the paired entropy factors among comparable positions within each trajectory while leaving support weights fixed. This tests whether the benefit of entropy weighting depends on where the weights are assigned (Appendix E.4).

Table 3: **Predictive entropy for supervision allocation.** Distillation uses positive-advantage trajectories and positively supported token positions. Step-200 reports ALFWorld success rate (%); Change reports the difference from 37.5% at step 150, in percentage points.

Token weighting	Unnormalized weight	Step-200	Change
GRPO only	–	36.7	−0.8
Support only	$s_{i,t}$	46.1	+8.6
+ Student uncertainty	$s_{i,t}u_{i,t}$	46.9	+9.4
+ Teacher certainty	$s_{i,t}v_{i,t}$	47.7	+10.2
Full weighting	$s_{i,t}u_{i,t}v_{i,t}$	**49.2**	**+11.7**
Shuffled entropy	$s_{i,t}u_{i,\pi(t)}v_{i,\pi(t)}$	45.3	+7.8

As shown in Table 3, combining student uncertainty and teacher certainty yields the largest gain, achieving a 49.2% success compared with 46.1% for support-only weighting. Shuffling the paired entropy factors reduces success to 45.3%. These results suggest that the benefit of entropy weighting depends on which which positions receive the weights.

5.3 Complementarity of the Two Selection Levels

Ablation of the two selection levels. We examine whether trajectory-level selection and token-level selection and weighting contribute complementary benefits. Starting from DL-OPSD, we remove either level or both in a 2×2 ablation, keeping all other settings fixed. This compares the full method with trajectory-only selection, token-only selection and weighting, and uniform distillation.

Table 4: **Ablation of the two selection levels.** Checkmarks indicate enabled components. The token-only and full configurations reuse the corresponding results in Table 2. The shaded row denotes DL-OPSD.

Trajectory-level	Token-level	ALFWorld	Search-QA	WebShop
✗	✗	60.2	40.0	77.3
✓	✗	64.8	39.8	74.2
✗	✓	69.5	36.7	68.8
✓	✓	**70.3**	**42.4**	**81.3**

Table 4 shows that neither level alone improves performance across all three environments, while their combination achieves the highest performance in each. On Search-QA and WebShop, either level alone reduces performance relative to uniform distillation, but combining them improves it. These results highlight the importance of jointly determining which trajectories receive distillation and how supervision is allocated within them.

Ablation of token-level factors. We further examine the contributions of teacher support, student uncertainty, and teacher certainty within positive-advantage trajectories. We remove each factor in turn while retaining the remaining components and the same training configuration.

Removing any factor lowers performance in all three environments (Table 5). Teacher support has the largest effect on ALFWorld, while student uncertainty has the largest effect on WebShop. In the latter environment, removing student uncertainty underperforms uniform weighting, indicating that teacher support and teacher certainty alone are insufficient to allocate supervision effectively in this setting. Together, the three factors account for both teacher guidance and the student's need for it.

Table 5: Ablation of the token-level weighting factors on positive-advantage trajectories. Uniform and Full reuse the corresponding configurations in Table 4. The shaded row denotes our full method.

Token weighting	ALFWorld	Search-QA	WebShop
Uniform	64.8	39.8	74.2
w/o positive support (s)	66.4	40.1	78.9
w/o student uncertainty (u)	68.8	40.1	71.9
w/o teacher certainty (v)	67.2	40.4	79.7
Full	**70.3**	**42.4**	**81.3**

6 Conclusion

We introduced DL-OPSD for skill internalization through selective auxiliary distillation at the trajectory and token levels. At comparable teacher–student gaps, we observed opposite signs of average original-action gain on positive- and negative-advantage trajectories. This suggests that trajectory outcomes provide an additional basis for selecting distillation supervision. DL-OPSD therefore restricts auxiliary distillation to positive-advantage trajectories while retaining reinforcement learning updates on all trajectories. Within selected trajectories, teacher support selects token positions, while student uncertainty and teacher certainty determine supervision weights. Experiments on ALFWorld, Search-QA, and WebShop demonstrate improved task performance without skill access at inference. Controlled experiments further show that trajectory selection improves gradient alignment between distillation and reinforcement learning, while token weighting uses student uncertainty and teacher certainty to further improve performance. These results suggest that execution feedback can guide the selection of self-guidance, helping language agents internalize external skills into their policies more effectively.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the implementation and execution of parts of the experiments, including code development and debugging. We also used generative AI tools to edit author-written text for grammar, clarity, and readability. Generative AI tools were not used to formulate the research ideas, hypotheses, methodology, or experimental design, nor were they used to generate synthetic datasets or formulate or prove mathematical claims. All AI-assisted code was reviewed and tested by the authors, all reported results were verified against the original experimental outputs, and all AI-assisted text was checked and revised by the authors. We take responsibility for the final content of this work, including all text, claims, code, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the environments, backbones, baselines, and training protocol in Section 4.1, and provide full hyperparameters, numerical bounds, and per-method objectives in the appendix.

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*, volume 2024, pp. 21246–21263, 2024.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

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.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026a.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, et al. Skillsbench: Benchmarking how well agent skills work across diverse tasks. *arXiv preprint arXiv:2602.12670*, 2026b.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, et al. Self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2605.15155*, 2026a.

Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Skill0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*, 2026b.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, et al. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents. *arXiv preprint arXiv:2604.10674*, 2026.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

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.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv e-prints*, pp. arXiv–2412, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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*, 2026.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

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.

Boyuan Zheng, Michael Y Fatemi, Xiaolong Jin, Zora Zhiruo Wang, Apurva Gandhi, Yueqi Song, Yu Gu, Jayanth Srinivasa, Gaowen Liu, Graham Neubig, et al. Skillweaver: Web agents can self-improve by discovering and honing skills. *arXiv preprint arXiv:2504.07079*, 2025.

APPENDIX CONTENTS

Appendix A: Related Work 12
Appendix B: Method Details 13
Appendix C: Experimental Details 13
Appendix D: Additional Results 15
Appendix E: Analysis Details 15

A RELATED WORK

A.1 SKILL AUGMENTATION AND INTERNALIZATION

Skills organize domain knowledge, operational procedures, and task strategies into reusable external guidance for language agents. SkillsBench systematically evaluates curated skills and shows that high-quality procedural knowledge can improve task performance across different models and agent frameworks (Li et al., 2026b). Skills can also be acquired from an agent's interaction experience. SkillWeaver distills web interactions into reusable operational interfaces and continually refines them through execution feedback (Zheng et al., 2025). SkillRL constructs a hierarchical skill library and integrates skill retrieval and recursive updating into reinforcement learning, allowing accumulated experience to continually support policy learning (Xia et al., 2026). These studies demonstrate the value of external skills for knowledge reuse and task execution. Using skills through context, however, requires agents to retrieve and read external guidance at inference time, introducing additional retrieval and context costs while making performance dependent on skill availability and retrieval quality. SKILL0 formulates skill internalization as a training objective and gradually withdraws skills during reinforcement learning, encouraging the policy to convert external guidance into its own capability and operate without skill access at inference time (Lu et al., 2026b).

A.2 ON-POLICY DISTILLATION AND SELF-DISTILLATION

On-policy distillation provides teacher supervision on sequences generated by the student, covering states visited by the student and reducing the distribution mismatch of offline distillation. Generalized Knowledge Distillation minimizes the teacher–student distribution difference on student-generated sequences, providing fine-grained token-level supervision for language models (Agarwal et al., 2024). Within this setting, token positions can differ in their learning value. TIP studies token importance through student predictive entropy and teacher–student divergence, showing that student uncertainty and teacher–student disagreement provide complementary signals for supervision selection (Xu et al., 2026). On-policy self-distillation further constructs the teacher and student from the same model under different contexts: the teacher receives privileged information available during training, while the student observes only the ordinary input and learns from the teacher on its own generated sequences. OPSD applies this paradigm using privileged information such as reference solutions (Zhao et al., 2026). For multi-turn agents, Skill-SD summarizes interaction trajectories into natural-language skills and provides them only to the teacher, creating training supervision for a student that acts without skills (Wang et al., 2026). Skills can therefore provide fine-grained guidance during training while the student retains the same input condition used at inference.

A.3 INTEGRATING RLVR WITH SELF-DISTILLATION

Reinforcement learning optimizes policies from environment outcomes, providing task-aligned feedback for agents, while trajectory-level outcomes offer limited information about individual decisions in long-horizon interaction. Self-distillation can complement this feedback with fine-grained predictive signals, motivating recent work on how the two sources of supervision should contribute to joint training. RLSD uses token-level prediction differences between a privileged teacher and the student to modulate policy update magnitudes, while environment outcomes determine the update direction (Yang et al., 2026). SDAR retains reinforcement learning as the primary objective and uses local teacher–student signals to softly gate an auxiliary self-distillation objective (Lu et al., 2026a). SRPO routes samples between group-relative policy optimization and self-distillation according to their task outcomes, with teacher entropy further modulating the distillation signal (Li et al., 2026a).

These studies combine execution feedback and self-guidance through update modulation, local gating, and sample routing.

B Method Details

B.1 Teacher Support Weighting

The teacher support weight uses

$$\phi(\Delta_{i,t}) = \min\left(\frac{[\Delta_{i,t}]_+}{\kappa_B + \epsilon}, 1\right), \qquad s_{i,t} = \mathbf{1}[\Delta_{i,t} > \delta]\, \phi(\Delta_{i,t}), \tag{11}$$

where $\delta = 10^{-3}$. We set κ_B to the 98th percentile of positive gaps in the current batch and use their maximum when fewer than 32 positive gaps are available. If the batch contains no positive gap, all support weights are set to zero.

B.2 Entropy Normalization

The student and teacher entropy factors use separate normalizations. Both entropies are computed over the full vocabulary; the Top-K approximation is used only for the distillation target. For the student, let κ_H be the 98th percentile of $H(p_{i,t})$ over valid policy-generated tokens in the current batch. We define

$$\widetilde{H}_S(p_{i,t}) = \min\left(\frac{H(p_{i,t})}{\kappa_H + \epsilon}, 1\right). \tag{12}$$

This batch-relative normalization adapts the student uncertainty scale during training while limiting it to $[0, 1]$. Teacher entropy is normalized by the maximum entropy over the vocabulary:

$$\widetilde{H}_T(q_{i,t}) = \mathrm{clip}\left(\frac{H(q_{i,t})}{\log|\mathcal{V}|}, 0, 1\right). \tag{13}$$

We use $\eta = 0.3$, so the student-uncertainty factor lies in $[1, 1.3]$.

B.3 Top-K Distillation Target

Let $S_{i,t}$ be the set of the K tokens with the highest teacher probability at position (i, t). The truncated teacher and student distributions used in the distillation target keep these K probabilities and collapse the remaining vocabulary into one residual mass,

$$\begin{aligned} \overline{q}_{i,t} &= \Big(\{q_{i,t}(v)\}_{v\in S_{i,t}},\ 1 - \textstyle\sum_{v\in S_{i,t}} q_{i,t}(v)\Big), \\ \overline{p}_{i,t} &= \Big(\{p_{i,t}(v)\}_{v\in S_{i,t}},\ 1 - \textstyle\sum_{v\in S_{i,t}} p_{i,t}(v)\Big). \end{aligned} \tag{14}$$

We compute the distillation term as $d_{i,t} = D_{\mathrm{KL}}(\overline{q}_{i,t} \| \overline{p}_{i,t})$. This forms a coarsened approximation to the full-vocabulary divergence while retaining the teacher's main candidate probabilities. We use $K = 8$.

C Experimental Details

C.1 Benchmarks

ALFWorld is a text-based household environment covering six task types (Pick, Look, Clean, Heat, Cool, and Pick2), where the agent selects executable actions from textual observations until it reaches the goal (Shridhar et al., 2020).

Search-QA follows the setup of Search-R1 (Jin et al., 2025), training on Natural Questions (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018) and evaluating on Natural Questions, TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023), HotpotQA, 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023). Natural Questions and HotpotQA are in-domain, while the remaining datasets are out-of-domain.

WebShop is an interactive online shopping environment where the agent retrieves products, browses pages, and completes a purchase according to a user instruction (Yao et al., 2022).

Table 6: Environment and DL-OPSD hyperparameters.

Setting	ALFWorld	Search-QA	WebShop
Tasks per update	16	128	16
Group size	8	8	8
Max. prompt length	2,048	4,096	4,096
Max. response length	512	512	512
Max. interaction turns	50	4	15
Validation interval	5	50	5
Reference KL coefficient	0.01	0.001	0.01
Distillation coefficient λ	0.001	0.001	0.001
Student-uncertainty coefficient η	0.3	0.3	0.3

C.2 Baselines

We compare against three categories of methods. (1) **Training-free methods.** Vanilla denotes the instruction model without task-specific post-training, while Skill-Prompt* retrieves and provides skills only at inference. (2) **Post-training methods.** GRPO (Shao et al., 2024) optimizes environment returns, and OPSD (Zhao et al., 2026) performs skill-conditioned on-policy self-distillation. Skill-GRPO provides retrieved skills during GRPO training and is evaluated either without skills or with skills as Skill-GRPO*. SKILL0 (Lu et al., 2026b) performs skill internalization through its in-context post-training objective. (3) **Hybrid methods.** GRPO+OPSD, RLSD (Yang et al., 2026), and SDAR (Lu et al., 2026a) combine reinforcement learning with supervision from a skill-conditioned teacher. A superscript * denotes evaluation with skills.

C.3 Evaluation Metrics

For ALFWorld, we report the overall task success rate and success rates for its six task types. Search-QA is evaluated by exact-match accuracy on seven question-answering datasets, with Avg. computed over all evaluation examples. WebShop reports task score (Score) and complete-task success rate (Acc.). All metrics are percentages, and higher values indicate better performance.

C.4 Training Configuration

We train on eight NVIDIA H800 GPUs using AdamW with a learning rate of 10^{-6} and a GRPO clipping parameter of 0.2. All trainable methods run for 300 policy updates and sample eight trajectories per task. We use the SkillRL skill library with keyword-matching retrieval for all privileged methods. Table 6 lists the environment-specific settings and the coefficients used by DL-OPSD.

C.5 Environment and Evaluation Details

ALFWorld follows the GiGPO(Feng et al., 2026) split and uses 128 fixed validation tasks. Search-QA uses Natural Questions and HotpotQA for training and evaluates 3,125 questions with an E5 retriever. WebShop uses the public 1,000-product subset and 128 validation tasks.

For each model and environment, we report the best validation checkpoint within the 300-update budget under the scheduled evaluation intervals. ALFWorld reports complete task success. WebShop reports task score and complete-task success. Search-QA uses exact-match accuracy, with its average weighted over 3,125 examples: 500 examples from each of the first six datasets and 125 from Bamboogle.

C.6 Baseline Objectives

We follow the baseline implementations released with the SDAR codebase. OPSD uses a token-level objective derived from the student-sampled reverse-KL term, with sigmoid weighting based on the teacher–student gap. GRPO+OPSD combines GRPO with a uniformly weighted version of this objective, while SDAR combines GRPO with sigmoid gap weighting. RLSD incorporates the

Table 7: **Comparison of training objectives.** The distillation objective identifies the form of the teacher-derived loss, while scope and token weighting specify how its supervision is allocated.

Method	Policy objective	Distillation objective	Scope	Token weighting
GRPO	GRPO	–	–	–
OPSD	–	Reverse-KL-derived	All	Sigmoid gap
GRPO+OPSD	GRPO	Reverse-KL-derived	All	Uniform
RLSD	Teacher-weighted GRPO	–	–	–
SDAR	GRPO	Reverse-KL-derived	All	Sigmoid gap
DL-OPSD	GRPO	Forward KL	$A_i > 0$	$\mathbf{1}[\Delta_{i,t} > \delta]\phi(\Delta_{i,t})u_{i,t}v_{i,t}$

Table 8: Multi-seed results on ALFWorld with Qwen3-1.7B. Values are mean $\pm$ standard deviation over training seeds.

Model	Method	ALFWorld
Qwen3-1.7B	GRPO+OPSD	60.67 ± 2.51
Qwen3-1.7B	SDAR	63.54 ± 5.12
Qwen3-1.7B	**DL-OPSD (Ours)**	71.61 ± 1.47

teacher signal into token-level policy advantages:

$$\widetilde{A}_{i,t} = A_i \left[(1-\rho) + \rho \, \mathrm{clip}(\exp(\mathrm{sign}(A_i)\Delta_{i,t}), 1-\epsilon_w, 1+\epsilon_w)\right], \tag{15}$$

Table 7 summarizes these differences.

D Additional Results

D.1 Multi-Seed Results

Table 8 reports three independent training runs for each method. We select the best validation checkpoint within the 300-update budget for each run, then compute the mean and standard deviation across the three selected results. The validation set and checkpoint-selection protocol follow Appendix C.5.

D.2 Efficiency and Computational Cost

Computational cost. We compare computational cost using one representative run of each method under the same WebShop configuration. All experiments use Qwen3-1.7B, eight NVIDIA H800 GPUs, 16 tasks per update, eight trajectories per task, and an actor micro-batch size of 8 per GPU. Timing results are averaged over 240 training updates between steps 1 and 300, excluding evaluation and checkpoint saving. Peak GPU memory is the maximum device-level memory usage observed by W&B across all eight GPUs over the complete run.

As shown in Table 9, DL-OPSD increases wall-clock time per update by 9.4% relative to GRPO. Its non-generation cost is comparable to GRPO+OPSD and SDAR, while its end-to-end update time is 37.7% and 38.7% lower, respectively, primarily due to lower generation time. Across all five methods, the observed peak device memory remains close to 79.1 GiB.

E Analysis Details

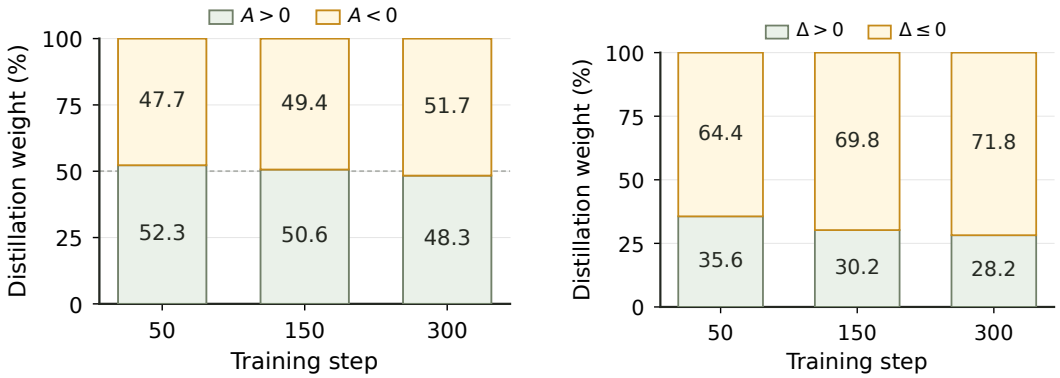
E.1 Distillation Allocation in Existing Methods

E.2 Continuation Analysis Protocol

We use decision states collected from the ALFWorld validation trajectories of the Qwen3-1.7B DL-OPSD checkpoint at step 300. At each state, the original action and student-sampled admissible alternatives are executed separately, after which the student completes the task without skills.

Table 9: Computational cost on WebShop with Qwen3-1.7B. All timing values are measured in seconds per update, and GPU memory is reported in GiB.

Method	Time/update	Generation	Teacher forward	Actor update	Non-generation	Peak memory
GRPO	240.3	217.1	–	15.4	23.2	79.17
GRPO+OPSD	421.9	379.4	14.7	18.3	42.5	79.14
RLSD	242.5	210.4	9.5	15.4	32.1	79.08
SDAR	429.2	387.5	14.6	17.8	41.8	79.09
DL-OPSD	263.0	221.5	13.2	20.2	41.5	79.18



(a) GRPO+OPSD on $A \neq 0$ trajectories

(b) SDAR over teacher–student gaps

Figure 5: **Auxiliary distillation weight allocation during Search-QA training.** (a) Within trajectories with nonzero group-relative advantage, uniform GRPO+OPSD distillation weight is divided between positive- and negative-advantage trajectories. Zero-advantage trajectories are excluded because their GRPO policy gradient is zero. (b) SDAR gate weight is divided between positions with positive and nonpositive teacher–student gaps. Each bar is normalized within its method, checkpoint, and analysis scope. DL-OPSD applies distillation only to trajectories with $A > 0$ and positions with $\Delta > \delta$.

Original-action gain is the success rate of continuations following the original action minus the mean success rate following its alternatives. The continuation randomness is paired across the compared actions and is independent of the rollout used to compute the trajectory advantage.

For this analysis, we compute the response-level teacher–student gap by averaging $\Delta_{i,t}$ over the policy-generated tokens in each decision response. We intersect the 5th–95th percentile gap ranges of the positive- and negative-advantage groups and retain the contiguous region where both groups have a Gaussian-kernel Kish effective sample size of at least 20. The resulting common interval $[0.0126, 0.0538]$ contains 50 positive-advantage and 56 negative-advantage states from 48 source-trajectory clusters. We compute 95% confidence intervals using 2,000 bootstrap resamples over source-trajectory clusters.

E.3 Random-Matched Trajectory Selection

The trajectory-selection diagnostic applies sigmoid gap weighting at every valid position. We map the teacher–student gap to a positive local score and normalize it within each trajectory:

$$\widehat{r}_{i,t} = \sigma(\beta \Delta_{i,t}), \qquad \widehat{w}_{i,t} = \frac{T_i \, \widehat{r}_{i,t}}{\sum_{j=1}^{T_i} \widehat{r}_{i,j}}, \tag{16}$$

where $\sigma(\cdot)$ is the sigmoid function and $\beta = 5.0$.

Let $\mathcal{P} = \{i : A_i > 0\}$ denote the positive-advantage set. For the random-matched control, we sample 64 candidate trajectory subsets, each containing $|\mathcal{P}|$ trajectories. We select the subset $\mathcal{R}$

whose total valid-token count is closest to that of $\mathcal{P}$:

$$\mathcal{R} \in \underset{\mathcal{R}_k,\, k=1,\ldots,64}{\arg\min} \left| \sum_{i \in \mathcal{R}_k} T_i - \sum_{i \in \mathcal{P}} T_i \right|, \qquad |\mathcal{R}_k| = |\mathcal{P}|. \tag{17}$$

The selected weights are rescaled to match the total normalized token weight of $\mathcal{P}$:

$$c = \frac{\sum_{i \in \mathcal{P}} \sum_{t=1}^{T_i} \widehat{w}_{i,t}}{\sum_{i \in \mathcal{R}} \sum_{t=1}^{T_i} \widehat{w}_{i,t}}, \qquad \widehat{w}_{i,t}^{\mathrm{Random}} = c\, \mathbf{1}[i \in \mathcal{R}] \widehat{w}_{i,t}. \tag{18}$$

If $\mathcal{P}$ is empty, auxiliary distillation is skipped for that batch. In the continued-training diagnostic, each variant computes these statistics from its current rollout batch. The selected trajectory count and total normalized token weight are matched exactly, and the total valid-token count is matched approximately.

E.4 Entropy-Factor Permutation

The shuffled-entropy control uses $e_{i,t} = u_{i,t} v_{i,t}$. Within each trajectory, positively supported positions are partitioned into four bins using the empirical quantiles of their positive teacher–student gaps. The paired factors are jointly permuted among positions in the same bin:

$$\widetilde{e}_{i,t} = e_{i,\pi_b(t)}, \qquad r_{i,t}^{\mathrm{shuffle}} = s_{i,t} \widetilde{e}_{i,t}, \tag{19}$$

where π_b is a random permutation within bin b. Bins containing fewer than two positions remain unchanged. The shuffled weights use the same trajectory-wise normalization as the other variants.

E.5 Gradient Analysis Protocol

Let $g_{\mathrm{RL}} = \nabla_\theta \mathcal{L}_{\mathrm{policy}}$ and $g_{\mathrm{distill}} = \nabla_\theta(\lambda \mathcal{L}_{\mathrm{distill}})$. We measure their gradient alignment over all trainable parameters as

$$C = \frac{\langle g_{\mathrm{RL}}, g_{\mathrm{distill}} \rangle}{\|g_{\mathrm{RL}}\|_2 \|g_{\mathrm{distill}}\|_2}. \tag{20}$$

The policy term uses the GRPO policy loss without reference KL, and the distillation term uses the full token-level rule, the Top-8 forward KL target, and $\lambda = 0.001$. Gradients are computed for the first decision response of each trajectory.

We analyze checkpoints at steps 25, 50, 75, 100, 125, and 150 from one ALFWorld GRPO run. At each checkpoint, 50 fixed task seeds generate groups of eight trajectories. We retain groups containing both successful and failed trajectories, yielding 83 groups in total. All-trajectory, random-matched, and positive-advantage gradients are computed from the same group. Random matching selects the same number of responses as positive-advantage selection and rescales their detached token weights to match its total normalized weight. We use 5,000 group-bootstrap resamples for the checkpoint curves and 10,000 paired group-bootstrap resamples for comparisons between random-matched and positive-advantage gradients.

E.6 Training-Dynamics Metrics

The success curves in Figure 3 report performance on the fixed validation task sets. The mean teacher–student gap averages $\log q_{i,t}(y_{i,t}) - \log p_{i,t}(y_{i,t})$ over all valid policy-generated response tokens before trajectory or positive-gap selection. KL to the reference is the token-averaged KL estimator used in the GRPO objective, and policy entropy is averaged over the same valid response tokens. Faint curves show the logged values. The displayed success curves use an exponential moving average with span 3, while the three step-level training metrics use span 8, corresponding to smoothing coefficients 0.5 and $2/9$, respectively.