SELF-DISTILLED AGENTIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has emerged as a central paradigm for post-training LLM agents, yet its trajectory-level reward signal provides only coarse supervision for long-horizon interaction. On-Policy Self-Distillation (OPSD) complements RL by introducing dense token-level guidance from a teacher branch augmented with privileged context. However, transferring OPSD to multi-turn agents proves problematic: compounding multi-turn instability destabilizes supervision, while skill-conditioned privileged guidance requires asymmetric treatment for negative teacher rejections may arise from imperfect skills retrieval or utilization. We introduce **SDAR** (**S**elf-**D**istilled **A**gentic **R**einforcement Learning), which treats OPSD as a gated auxiliary objective while keeping RL as the primary optimization backbone. SDAR maps detached token-level signals into a sigmoid gate, strengthening distillation on teacher-endorsed positive-gap tokens and softly attenuating negative teacher rejections. Across the Qwen2.5 and Qwen3 families on ALFWorld, WebShop, and Search-QA, SDAR substantially improves over GRPO (up to $+9.4\%$ on ALFWorld and $+8.7\%$ on Search-QA on Qwen2.5-3B, and $+10.2\%$ on WebShop-Acc on Qwen2.5-7B), stabilizes training on small backbones where naive GRPO+OPSD collapses, and matches or outperforms hybrid RL–OPSD baselines in the majority of settings across model scales. Our code is available at `https://anonymous.4open.science/r/SDAR-2B07/`.

1 INTRODUCTION

Agentic post-training has become a central challenge for Large Language Models (LLMs) (Guo et al., 2025; Team et al., 2025; Yang et al., 2025; Comanici et al., 2025; Team et al., 2026b). Unlike static single-turn reasoning, multi-turn agents interact with environments over extended horizons, where each action changes future observations and each generated response becomes part of the context for subsequent decisions (Shen et al., 2023; Shi et al., 2025; Jimenez et al., 2023).

Two paradigms naturally emerge as complementary forces: Reinforcement Learning (RL) (Shao et al., 2024; Dong et al., 2025; Feng et al., 2025) provides task-level optimization grounded in environment or verifier feedback, whereas On-Policy Distillation (OPD) (Ye et al., 2026; Yang et al., 2026b; Team et al., 2026a; GLM-5-Team et al., 2026) and On-Policy Self-Distillation (OPSD) (Zhao et al., 2026; He et al., 2026; Zhang et al., 2026) provide dense token-level guidance from a teacher branch. Yet, OPSD does not transfer cleanly to multi-turn agent training:

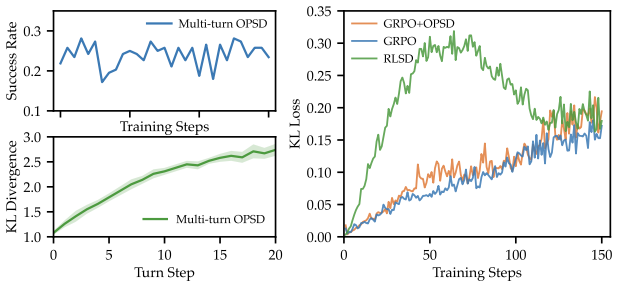


Figure 1: **Left**: Multi-turn OPSD Instability, with performance and KL reported. **Right**: RLSD-Style Instability.

[Observation-1] Multi-turn OPSD Instability Once the student agent inevitably drifts from the teacher support, the once-helpful token-level supervision becomes increasingly unreliable. This

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

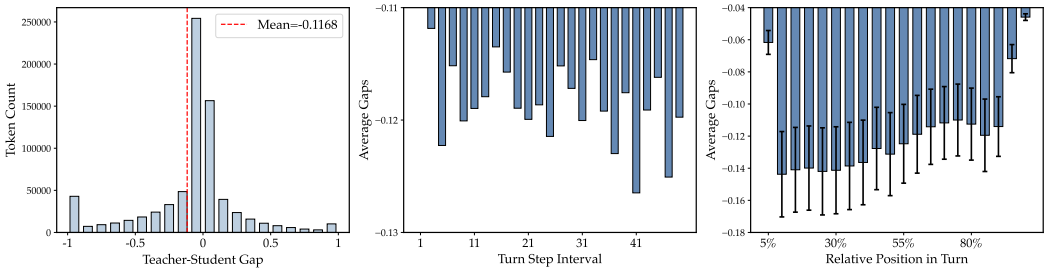


Figure 2: **Preliminary Study. Left**: Token count distribution partitioned by Teacher-Student gap. **Middle**: Average teacher-student gap indexed by turn step. **Right**: Average gap indexed by relative position within a turn.

compounding error leads to surging per-turn KL divergence and catastrophic degradation in task performance, as shown in Figure 1. TCOD (Wang et al., 2026b) attempts to address this through curriculum learning, but relies on rigid temporal schedules or trajectory-depth thresholds.

[Observation-2] Asymmetric Trust in Privileged Guidance In OPSD, the teacher is not an independently stronger model but the same policy augmented with privileged training-only context (e.g., retrieved skills), making its token-level guidance inherently asymmetric (Penaloza et al., 2026). When the privileged teacher assigns a *higher* probability to a student-sampled token y_t, the skill provides a positive endorsement of an on-policy behavior that the student can generate but has not fully internalized—such signals are well-suited for distillation.

Conversely, when the teacher assigns a *lower* probability, the signal warrants caution: a negative gap may reflect a genuinely undesirable token, but in skill-conditioned OPSD it can also stem from instability of the privileged context: **(1) Skill Quality.** Retrieved skills may be irrelevant, incomplete, or redundant. **(2) Skill Utilization.** The teacher may fail to ground even relevant skills into reliable token-level preferences (Chen et al., 2019). **(3) Multi-turn Drift.** As trajectories unfold, the teacher-student gap tends to widen across turns (Figure 2, Middle), amplifying early mismatches over successive decisions (Ross et al., 2011). Our preliminary study on Qwen2.5-3B-Instruct shows that negative-gap tokens exceed 50% of all tokens (Figure 2), making this issue pervasive.

A stark realization emerges: for multi-turn agents, RL should reign as the primary optimization backbone, while OPSD is relegated to a carefully controlled auxiliary role.

But how should this auxiliary role be controlled? RLSD (Yang et al., 2026a) directly uses self-divergence to re-weight token-level RL advantages, but can substantially amplify updates early in training (see Figure 1, Right).

We take a different path: the OPSD loss is treated as a direct, auxiliary optimization objective, leaving the verifier-driven RL policy loss untouched and thereby strictly preserving the semantics and unbiasedness of the RL advantage. To overcome instability of multi-turn OPSD and privileged guidance, distillation is not performed uniformly on every token. Instead, tokens are selectively distilled via an adaptive, smooth gating mechanism rather than a hand-crafted, rigid schedule (such as Skill-SD (Wang et al., 2026a) and HDPO (Ding, 2026)). Inspired by TIP (Xu et al., 2026), we use token-level signals (such as student entropy or teacher-student divergence) to control the gate's activation. The core philosophy is simple: *let each token decide the intensity of its own supervision.* This yields a dynamic curriculum operating at the finest granularity: the token level. We trust positive teacher endorsements more strongly, while applying negative teacher rejections more conservatively.

We validated our method across the Qwen2.5 and Qwen3 model families on three diverse benchmarks for llm-based agents: ALFWorld (Shridhar et al., 2020), WebShop (Yao et al., 2022), and Search-QA (Jin et al., 2025). SDAR achieves substantial improvements over GRPO ($+9.4\%$ on ALFWorld and $+8.7\%$ on Search-QA on Qwen2.5-3B, and $+10.2\%$ on WebShop-Acc on Qwen2.5-7B), and matches or outperforms RL–OPSD hybrid methods such as Skill-SD and RLSD in most settings. Our contributions are listed below,

- We identify multi-turn OPSD instability and asymmetric trust in privileged guidance as two key barriers to combining OPSD with RL.
- We propose SDAR, a sigmoid-gated distillation mechanism that amplifies positive teacher endorsements and attenuates negative rejections, serving as a stable auxiliary to RL.
- Experiments on three agent domains show SDAR's strong performance, with graceful degradation under noisy skill retrieval.

2 Related Work

2.1 Agentic Post-Training

Reinforcement learning with verifiable rewards has proven effective for LLM reasoning (Shao et al., 2024; Guo et al., 2025) and is increasingly applied to autonomous agents in dynamic environments (Ye et al., 2025; Shridhar et al., 2020; Wang et al., 2023), where sequential decision-making demands agentic RL as a core post-training recipe (Dong et al., 2025; Feng et al., 2025). Complementarily, On-Policy Distillation (OPD) (Agarwal et al., 2024; Gu et al., 2026) provides dense token-level supervision on student-generated trajectories: GKD-style methods (Agarwal et al., 2024; Wen et al., 2023) minimize token-level divergences, while PG-style methods (Yang et al., 2026a; Xu et al., 2026) convert discrepancy into token-level rewards. On-Policy Self-Distillation (OPSD) (Zhao et al., 2026; He et al., 2026) further removes the need for a separate teacher by injecting privileged context.

2.2 Hybrid Methods

Recent works combine RL with distillation to leverage their complementary strengths. Skill-SD (Wang et al., 2026a) and HDPO (Ding, 2026) employ hand-crafted schedules to mix the two objectives, while RLSD (Yang et al., 2026a) directly re-weights RL advantages with self-divergence, risking unstable early-training amplification. These approaches either rely on rigid heuristics or compromise the unbiasedness of RL gradients.

3 Method

3.1 Problem Setup

We consider a multi-turn agent that interacts with an environment over a finite horizon. Given an initial prompt or task description x, at turn k the agent receives an observation o_k, generates a response a_k, and the environment returns the next observation o_{k+1}. Each response a_k may contain both intermediate reasoning tokens and executable action tokens. For notational simplicity, we flatten all valid response tokens in one trajectory into a single token sequence

$$y = (y_1, \ldots, y_T) \sim \pi_\theta(\cdot \mid x),$$

where π_θ denotes the student policy and T is the total number of valid response tokens. At token position t, we denote the self-student context by

$$s_t = (x, y_{<t}),$$

and the self-teacher context by

$$s_t^+ = (x, c^+, y_{<t}),$$

where c^+ denotes privileged training-only context available only to the teacher branch, such as reference answers, skills (ours), or other auxiliary information not accessible at test time.

Skills Retrieval We retrieve task-relevant *skills*—compact, structured demonstrations that encode domain-specific knowledge such as sub-goal decompositions or action templates. We implement four retrieval strategies of varying quality to evaluate the robustness of our framework to the fidelity of the retrieved context: **(1) UCB Retrieval**, **(2) Keyword Matching (KM)**, **(3) Full Retrieval**, and **(4) Random Retrieval**. In addition, we include a skill-content control, **Empty**, which supplies the teacher with an empty skill context and isolates the effect of real skill content from any generic distillation regularization.

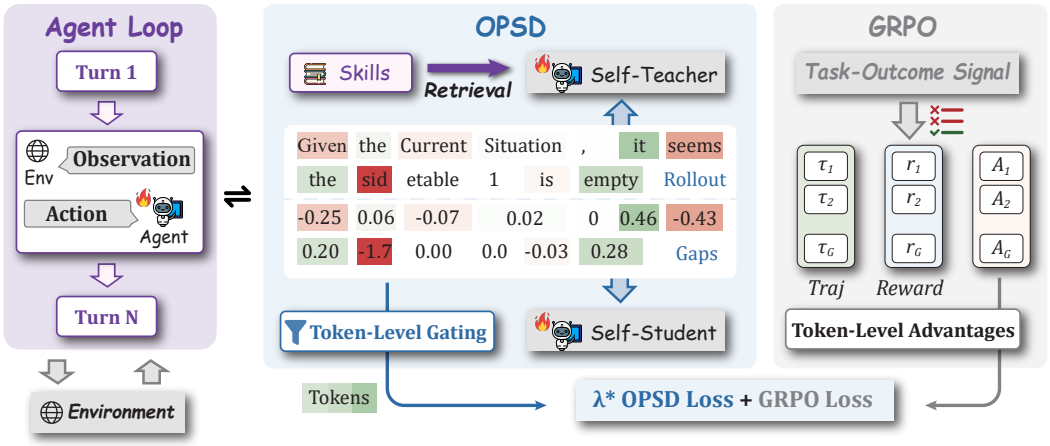


Figure 3: **Illustrations of SDAR,** which trains multi-turn agents using token-level OPSD loss and GRPO loss.

Skill retrieval is cast as a multi-armed bandit problem over the skill library $\mathcal{E} = \{e_1, \ldots, e_M\}$. For each incoming task, **UCB Retrieval** selects the single highest-scoring skill file according to the Upper Confidence Bound (UCB) criterion:

$$\text{score}(e) \;=\; \bar{r}(e) \;+\; c\sqrt{\frac{\ln N_{\text{ucb}}}{n(e)}}, \tag{1}$$

where $\bar{r}(e)$ is the running mean reward obtained when skill e was previously supplied as context, N_{ucb} is the total number of retrieval queries issued for the same task type, $n(e)$ is the number of times e has been selected, and c controls the exploration–exploitation trade-off. **KM** bypasses the bandit formulation and instead identifies the task scenario by matching keywords in the task description against predefined category labels, directly retrieving skill files associated with the matched category.

3.2 Optimization Goals

Our method is designed as an auxiliary objective on top of a standard policy optimization GRPO loss. The overall training objective is

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{GRPO}}(\theta) + \lambda_{\text{SDAR}} \cdot \mathcal{L}_{\text{SDAR}}(\theta),$$

where $\mathcal{L}_{\text{GRPO}}$ is the original policy loss and $\mathcal{L}_{\text{SDAR}}$ is our on-policy self-distillation objective.

Let $m_t \in \{0, 1\}$ be the response mask indicating whether token t is valid. We define masked token averaging as

$$\text{Agg}(z_{1:T}) = \frac{\sum_{t=1}^{T} m_t z_t}{\sum_{t=1}^{T} m_t}.$$

RL Optimization For each input x, GRPO samples a group of responses

$$\{y^{(i)}\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x),$$

and computes a sequence-level advantage $A^{(i)}$ from environment rewards. Using a reference policy π_{ref}, the GRPO objective can be written as

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G} \text{Agg}\Big(\min\Big(r_t^{(i)} A^{(i)},\;\; \text{clip}(r_t^{(i)}, 1-\epsilon, 1+\epsilon) A^{(i)}\Big)\Big) \;+\; \beta \cdot D_{\text{KL}}^{\text{avg}},$$

where $r_t^{(i)} = \pi_\theta(y_t^{(i)} \mid s_t^{(i)}) / \pi_{\theta_{\text{old}}}(y_t^{(i)} \mid s_t^{(i)})$ is the importance sampling ratio, and $D_{\text{KL}}^{\text{avg}} = \frac{1}{G}\sum_{i=1}^{G} \text{Agg}\Big(D_{\text{KL}}\big(\pi_\theta(\cdot \mid s_t^{(i)}) \,\|\, \pi_{\text{ref}}(\cdot \mid s_t^{(i)})\big)\Big)$.

OPSD Optimization At a fixed token position t, the teacher and student induce conditional token distributions $\pi_T(\cdot \mid s_t^+)$ and $\pi_\theta(\cdot \mid s_t)$, respectively. The per-token reverse KL divergence is defined as:

$$D_{\mathrm{RKL}}^{(t)} = D_{\mathrm{KL}}\big(\pi_\theta(\cdot \mid s_t) \,\big\|\, \pi_T(\cdot \mid s_t^+)\big) = \sum_{v \in \mathcal{V}} \pi_\theta(v \mid s_t) \log \frac{\pi_\theta(v \mid s_t)}{\pi_T(v \mid s_t^+)}.$$

To efficiently derive an *importance signal* without computing the expensive full-vocabulary summation, we take a single-sample estimate on the student-sampled token $y_t \sim \pi_\theta(\cdot \mid s_t)$. The negation of this estimate directly yields the Teacher-Student log-probability gap Δ_t:

$$\Delta_t = -\widehat{D}_{\mathrm{RKL}}^{(t)} = \log \pi_T(y_t \mid s_t^+) - \log \pi_\theta(y_t \mid s_t).$$

3.3 Token-Level Gating

The key idea is to convert privileged teacher guidance into a token-level trust weight, while keeping the verifier-driven RL objective unchanged. We introduce a token-level gate $g_t \in [0, 1]$ that modulates the OPSD signal on each student-sampled token, and apply it to a sampled-token surrogate so that different gating strategies share the same optimization. Let

$$\Delta_t \;=\; \mathrm{sg}\Big(\log \pi_\theta^+(y_t \mid s_t^+) \;-\; \log \pi_\theta(y_t \mid s_t)\Big)$$

denote the *detached* Teacher-Student log-probability gap on the student-sampled token,

$$h_t = -\sum_{v \in \mathcal{V}} \pi_\theta(v \mid s_t)\, \log \pi_\theta(v \mid s_t)$$

denote the student entropy at position t. We compose each raw score with the logistic sigmoid σ so that every gate is smooth, differentiable, and naturally bounded in $(0, 1)$. The sharpness parameter $\beta > 0$ controls the transition between conservative attenuation and strong activation.

We instantiate three gating strategies:

1. *Entropy gating*: $g_t = \sigma(\beta\, h_t)$, targeting high-entropy positions where student is uncertain.
2. *Gap gating*: $g_t = \sigma(\beta\, \Delta_t)$, assigning larger weights to positive-gap tokens endorsed by teacher while attenuating negative-gap tokens.
3. *Soft-OR gating*: $g_t = \sigma\big(\beta\big[1 - (1 - h_t)(1 - \Delta_t)\big]\big)$, combining student uncertainty and teacher-student gap.

In all cases, the gate is detached via $\mathrm{sg}(\cdot)$, so gradients flow exclusively through the student log-probability. The token-level loss is

$$\ell_t^{\mathrm{SDAR}} \;=\; g_t \cdot \big(\log \pi_\theta^+(y_t \mid s_t^+) - \log \pi_\theta(y_t \mid s_t)\big), \mathcal{L}_{\mathrm{SDAR}} \;=\; \mathrm{Agg}\big(\ell_t^{\mathrm{SDAR}}\big).$$

With gap gating, the sigmoid gate implements asymmetric token-level modulation: positive-gap tokens receive stronger auxiliary distillation, while negative-gap tokens are softly attenuated. We also provide theoretical analysis in Appendix A.

4 Experiment

4.1 Experiment Setup

Benchmarks We evaluate our methods on ALFWorld (Shridhar et al., 2020), Search-based QA (Jin et al., 2025), and Webshop (Yao et al., 2022). *ALFWorld* is a text-based game aligned with the ALFRED embodied AI benchmark, including 3,827 task instances across six categories of household activities: Pick, Look, Clean, Heat, Cool, and Pick2. *Search-based QA* contains several widely-used search-augmented QA benchmarks, including single-hop QA datasets (NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023)) and multi-hop QA datasets (HotpotQA (Yang et al., 2018), 2Wiki (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023)). *WebShop* is a complex, web-based interactive environment designed to test the LLM agents in realistic online shopping scenarios. Agents navigate a realistic web interface to find and purchase products matching user specifications. We select 128 fixed tasks in validation set, aligning with Feng et al. (2025).

Table 1: **Performance on ALFWorld, Search-QA and WebShop tasks.** We report the success rate (%) on ALFWorld, accuracy (%) on Search-QA, and Score/Acc (%) on WebShop (128 tasks). * means validation with skills. **Best** and second-best are highlighted.

	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	44.4	11.1	6.2	15.4	28.6	12.5	21.9	24.6	48.1	31.0	26.3	25.3	7.2	59.7	31.7	6.7	0.8
Skill-Prompt*	51.7	66.7	48.4	0.0	4.3	10.0	28.9	23.7	46.2	30.6	24.4	22.1	7.5	12.5	23.9	0.2	0.8
OPSD	48.8	41.7	16.7	0.0	15.8	16.7	28.1	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0	11.3	3.1
GRPO	91.2	62.5	96.2	61.9	65.0	47.4	75.0	39.3	60.6	41.1	37.4	34.6	15.4	26.4	36.4	79.8	63.3
Skill-GRPO	88.9	71.4	58.8	70.6	40.7	29.2	60.2	43.5	58.8	43.0	36.8	32.2	11.7	12.5	34.1	77.3	60.9
Skill-GRPO*	94.3	57.1	**100**	66.7	73.1	57.1	80.5	44.3	59.6	44.3	39.0	36.1	14.5	14.9	36.1	76.3	66.4
GRPO+OPSD	**100**	**82.4**	85.7	**75.0**	70.0	60.0	81.2	44.9	61.2	**45.2**	40.4	38.5	16.0	66.1	44.6	77.8	66.4
Skill-SD	88.2	50.0	96.2	52.4	65.0	57.9	73.4	44.4	60.4	44.0	39.5	**40.4**	15.4	64.9	44.1	75.9	64.0
RLSD	87.9	75.0	90.9	**75.0**	73.1	68.4	79.7	41.5	58.6	42.3	40.4	40.2	**16.8**	**66.9**	43.8	84.4	66.4
SDAR	97.1	62.5	**100**	61.9	**75.0**	**84.2**	**84.4**	**45.8**	**61.6**	45.0	**40.9**	40.2	16.2	66.1	**45.1**	**85.0**	**68.0**
Qwen2.5-7B-Instruct																	
Vanilla	36.1	22.2	3.1	0.0	0.0	0.0	12.5	25.2	50.8	29.5	29.0	29.0	10.4	63.7	33.9	5.9	1.6
Skill-Prompt*	51.7	50.0	32.3	5.3	4.3	0.0	23.4	30.9	52.1	32.7	32.7	27.9	12.7	66.1	36.4	1.7	0.8
OPSD	50.0	60.0	22.7	21.4	17.6	9.5	32.8	8.8	8.6	17.5	2.5	4.2	0.5	1.2	6.2	4.5	2.3
GRPO	91.2	87.5	96.2	81.0	65.0	57.9	81.2	45.1	63.7	44.0	43.6	43.2	16.8	37.6	42.0	80.9	72.6
Skill-GRPO	88.5	66.7	65.2	61.1	57.7	73.1	69.5	45.2	63.7	45.7	43.1	43.3	19.6	21.4	40.3	80.4	71.9
Skill-GRPO*	**100**	83.3	96.4	83.3	75.0	**78.9**	**88.3**	44.8	63.0	45.1	43.7	43.7	20.5	71.4	47.5	87.0	81.2
GRPO+OPSD	91.4	61.5	**100**	87.5	76.5	52.2	80.4	**47.3**	**64.5**	46.9	43.8	39.3	18.0	69.4	47.0	86.8	76.5
Skill-SD	93.9	**93.8**	90.9	**100**	69.2	68.4	85.1	47.1	**64.5**	47.8	44.2	42.1	20.2	69.0	47.8	86.1	76.5
RLSD	**100**	87.5	92.3	58.8	**80.0**	65.2	82.0	46.8	63.0	44.4	**45.5**	**48.9**	**21.5**	73.0	49.0	87.4	77.3
SDAR	94.7	75.0	**100**	86.7	68.2	**78.9**	85.9	46.3	63.5	**48.2**	43.8	48.4	19.6	**73.0**	**49.0**	**89.4**	**82.8**
Qwen3-1.7B-Instruct																	
Vanilla	25.0	22.2	3.1	0.0	21.4	4.2	12.5	29.4	46.9	37.0	23.5	19.6	6.4	10.5	24.8	46.5	4.7
Skill-Prompt*	10.3	50.0	16.1	0.0	0.0	5.0	9.4	29.4	46.5	36.2	22.9	20.8	4.3	10.1	24.3	23.0	2.3
OPSD	26.3	33.3	9.1	0.0	4.5	5.3	14.1	4.2	8.3	4.6	6.6	15.3	0.7	1.2	5.8	47.4	9.3
GRPO	71.1	41.7	36.4	40.0	31.8	31.6	46.1	40.0	**58.9**	43.5	35.4	30.3	12.0	65.7	40.8	67.3	38.3
Skill-GRPO	27.6	**54.5**	22.7	27.3	0.0	19.2	21.1	39.2	58.6	43.9	35.2	28.2	11.5	66.1	40.4	73.4	46.1
Skill-GRPO*	31.4	42.9	51.9	8.3	11.5	7.1	28.1	38.0	58.4	43.9	36.3	29.0	12.5	**66.9**	40.7	80.4	50.0
GRPO+OPSD	38.2	50.0	30.8	28.6	30.0	21.1	32.0	**40.7**	**58.9**	45.0	**37.0**	34.6	**13.3**	65.7	**42.2**	70.7	38.3
Skill-SD	52.9	37.5	69.2	**42.9**	**60.0**	**36.8**	52.3	39.1	57.5	**45.4**	34.8	34.1	10.7	64.1	40.8	**81.8**	53.9
RLSD	50.0	37.5	61.5	19.0	50.0	21.1	42.2	38.6	57.3	43.0	34.5	34.1	11.5	65.3	40.6	74.0	50.8
SDAR	**73.5**	25.0	**76.9**	33.3	40.0	**36.8**	**53.9**	39.7	**58.9**	45.3	35.9	**35.5**	12.6	65.3	41.9	76.8	**58.6**

Implementation Details. We train the Qwen2.5-Instruct and Qwen3-Instruct series using SDAR for 150 steps on 8 H800 GPUs. For ALFWorld, we adopt the training data split from GiGPO (Feng et al., 2025), with each batch sampling 16 tasks and 8 rollouts per prompt, and a maximum prompt length of 2,048 tokens. For Search-QA, we follow the experimental setup of Search-R1 (Jin et al., 2025), using E5 (Wang et al., 2022) as the retriever. The training data are drawn from NQ and HotpotQA, making these two benchmarks in-domain, while the remaining datasets serve as out-of-domain evaluation. Each batch samples 128 tasks with a maximum prompt length of 4,096 tokens. For Webshop, 1000 tasks are selected for training, with each batch sampling 16 tasks and 8 rollouts per prompt, and a maximum prompt length of 4,096 tokens. We set the `SkillBank` from SkillRL (Xia et al., 2026) for all three environments. We set $\lambda_{\text{SDAR}} = 0.01$ and $\beta = 5.0$ in our experiments.

Baselines We compare SDAR against three categories of methods on three base models. **(1) Training-free methods.** *Skill-Prompt* retrieves task-relevant skills from the `SkillBank` via keyword matching (KM) and prepends them to the input prompt at inference time. **(2) Post-training methods,** such as GRPO (Shao et al., 2024), OPSD (Zhao et al., 2026) and Skill-GRPO. *Skill-GRPO* augments GRPO by retrieving skills via KM and injecting them into the training prompt; at test time it can run with (*Skill-GRPO**) or without retrieved skills. **(3) Hybrid methods**, that combine RL with privileged knowledge distillation, such as GRPO+OPSD, and Skill-SD (Wang et al., 2026a), RLSD (Yang et al., 2026a). *GRPO+OPSD* simply adds the OPSD distillation loss as an auxiliary objective on top of GRPO training. All the algorithms of SDAR and other baselines are detailed in Appendix A.

4.2 Main Results

Overall Performance. As summarized in Table 1, SDAR achieves the best or second-best results in the majority of settings. Compared to GRPO, it delivers substantial gains across scales: on Qwen2.5-3B, ALFWorld improves by $+9.4$ (84.4 vs. 75.0) and Search-QA-Avg by $+8.7$; on Qwen2.5-7B,

WebShop-Acc improves by $+10.2$ (82.8 vs. 72.6). Against the strongest hybrid baselines, SDAR attains the best Qwen2.5-3B ALFWorld (84.4) and Search-QA-Avg (45.1, over GRPO+OPSD's 44.6 and Skill-SD's 44.1), and matches or leads on Qwen2.5-7B Search-QA-Avg; the margins on Search-QA are modest, and the full per-subset breakdown is in Table 1. Standalone OPSD collapses across all three backbones (near-zero on Search-QA), confirming that pure privileged distillation is not directly usable in multi-turn agentic RL. The naive GRPO+OPSD combination remains competitive with GRPO on Qwen2.5-3B/7B ALFWorld but degrades sharply on the smallest backbone (Qwen3-1.7B ALFWorld: 32.0 vs. 46.1), where unbounded distillation gradients overwhelm the RL signal. This is precisely where our token-level gating is most necessary: SDAR recovers to 53.9 on the same setting, a $+21.9$ gain over GRPO+OPSD and the largest improvement across all settings.

Skills Internalization. Beyond overall performance, SDAR successfully *internalizes* privileged knowledge rather than superficially relying on it at inference (Lu et al., 2026). While Skill-GRPO shows a massive performance drop when tested without skills (e.g., 60.2 vs. 80.5 on ALFWorld-3B) and even underperforms vanilla GRPO due to harmful distributional dependencies, SDAR requires no external skills during inference. Yet, it surpasses even the skill-augmented Skill-GRPO* in most settings, achieving 84.4 on ALFWorld-3B and a striking 53.9 (vs. 28.1) on ALFWorld-1.7B. These consistent gains confirm that our token-level gated distillation genuinely transfers underlying knowledge into the policy's parameters.

Strong Generalization. SDAR also generalizes better than the hybrid baselines GRPO+OPSD, Skill-SD, and RLSD. On Qwen2.5-3B, it reaches 84.4% on ALFWorld, ahead of GRPO+OPSD (81.2%), RLSD (79.7%), and Skill-SD (73.4%), and achieves the best WebShop accuracy of 68.0%, compared with 66.4% for both GRPO+OPSD and RLSD and 64.0% for Skill-SD. The advantage widens on the smaller Qwen3-1.7B, which may struggle to utilize retrieved skills effectively. Here, GRPO+OPSD collapses to 32.0% on ALFWorld, well below GRPO's 46.1%, Skill-GRPO drops further to 21.1%, and RLSD reaches only 42.2%. SDAR instead achieves the highest score of 53.9%, showing that our gating mechanism incorporates privileged knowledge more robustly without sacrificing generalization.

4.3 Training Dynamics

To understand how SDAR adapts during RL optimization, we track two metrics for Qwen2.5-7B on ALFWorld (Figure 5). *(a)* The mean teacher-student log-probability gap $\bar{\Delta} = \mathbb{E}_t[\Delta_t]$ stays negative throughout training: on average, the privileged teacher assigns lower probability than the student to sampled tokens, a regime in which naïve distillation would degrade performance. Meanwhile, $\bar{\Delta}$ steadily approaches zero, consistent with the gate concentrating distillation on the tokens where the teacher *does* help. Figure 4 supports this: SDAR closes the teacher-student gap faster than naïve GRPO+OPSD while also reaching a higher success rate, indicating that selective gating transfers useful signals more efficiently without the instability of uniform distillation. *(b)* The gate activation ratio, i.e., the fraction of tokens with $g_t > 0.5$, stays below 0.5 for most of early training, so most negative-gap tokens are suppressed. As the student improves, the ratio rises, reflecting that more tokens enter the regime of constructive teacher guidance.

4.4 Robustness Analysis

To test whether SDAR depends on high-quality skill retrieval, we fix the best configuration ($\lambda = 0.01$, $\beta = 5.0$) and evaluate four retrieval strategies of increasing quality, together with an **Empty** control that gives the teacher an empty skill context (Table 2). The Empty control separates the contribution of skill *content* from any generic regularization effect of distillation: it stays at the *w/o OPSD* baseline and even drops slightly on ALFWorld (-0.5). In contrast, all four retrieval strategies, each supplying a real but possibly noisy skill, outperform pure GRPO. Even **Random Retrieval**, which selects skills with no task awareness, gains $+1.9$ on ALFWorld ($+2.4$ over Empty), showing that the improvement comes from skill content rather than from regularization alone. Better retrieval amplifies the gain: **Keyword Matching** reaches $+4.7$ on ALFWorld and $+10.2$ on WebShop-Acc, surpassing UCB on WebShop, and performance rises monotonically from Empty to Random, Full, and KM/UCB, most sharply on WebShop-Acc. These results show that SDAR transfers genuine skill knowledge

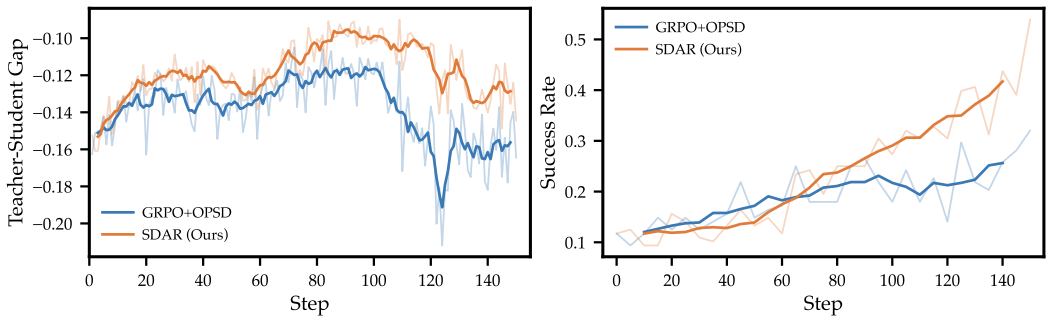


Figure 4: **Comparisons with naive GRPO+OPSD.** Average teacher-student gap (Left) and Success Rate (Right) during the training of Qwen3-1.7B-Instruct.

Table 2: Robust Testing of different skill retrieval methods of Qwen2.5-7B-Instruct. Web refers to WebShop.

Method	ALFWorld	Web-Score	Web-Acc
UCB	86.8$_{+5.6}$	87.5$_{+6.6}$	81.2$_{+8.6}$
KM	85.9$_{+4.7}$	89.4$_{+8.5}$	82.8$_{+10.2}$
Full	83.2$_{+2.0}$	87.2$_{+6.3}$	78.1$_{+5.5}$
Random	83.1$_{+1.9}$	82.5$_{+1.6}$	73.6$_{+1.0}$
Empty	80.7$_{-0.5}$	81.4$_{+0.5}$	73.4$_{+0.8}$
w/o OPSD	81.2	80.9	72.6

Table 3: Ablations of different gating mechanisms of Qwen2.5-7B-Instruct. WebShop reports task accuracy.

Method	ALFWorld	WebShop
Soft Gating (Ours)	85.9 $_{+5.5}$	82.8 $_{+6.3}$
Neg-Gap Only	84.7 $_{+4.3}$	81.2 $_{+4.7}$
Pos-Gap Only	82.6 $_{+2.2}$	78.1 $_{+1.6}$
Reverse Soft Gating	76.2 $_{-4.2}$	66.4 $_{-10.1}$
Hard Gating	81.5 $_{+1.1}$	79.7 $_{+3.2}$
Reverse Hard Gating	74.8 $_{-5.6}$	68.7 $_{-7.8}$
w/o Gating	80.4	76.5

rather than acting as a generic regularizer, and that its token-level gating keeps the gain robust across retrieval quality by retaining positive endorsements and attenuating unreliable rejections.

4.5 Ablation Studies

Gating Mechanism Design. Table 3 compares our sigmoid gate (*Soft*) with binary token filtering (*Hard*), *Reverse* variants that suppress positive-gap tokens and amplify negative-gap ones, and two half-gates that act in only one direction: *Neg-Gap Only* attenuates negative-gap tokens, and *Pos-Gap Only* amplifies positive-gap tokens. Soft gating outperforms hard gating, as smooth weighting retains partial credit that binary selection discards. Both Reverse variants fall below the ungated baseline, supporting our principle of trusting positive endorsements while attenuating negative rejections. Each half-gate improves over the ungated baseline on its own (roughly $+4$ for Neg-Gap Only and $+2$ for Pos-Gap Only), and their gains together approximately match the full gate, indicating that the two directions are complementary. Negative-gap suppression contributes more because negative gaps dominate early training: the mean gap stays negative and the gate-activation ratio remains below 0.5 (Figure 5). The benefit of positive-gap boosting grows as the student matures and more tokens enter the positive-gap regime.

Token-Level Gating Strategy. As shown in Figure 6, Teacher-Student Gap gating consistently outperforms both the entropy and soft-OR gating strategies (introduced in Section 3.3), achieving a higher asymptotic success rate ($\sim$0.84) and a steeper performance climb after the initial 100 steps. We attribute this superiority to the directness of the Teacher-Student gap (Δ_t) as an importance signal, which precisely measures the teacher's disagreement with the student's chosen token. In contrast, entropy (h_t) acts as an indirect proxy that may erroneously activate on uncertain but already well-handled tokens, while soft-OR dilutes the gating signal by triggering when only one score is moderately large, thereby reducing its selectivity. All remaining experiments default to gap gating.

Sharpness β. Figure 7 evaluates the impact of sigmoid sharpness across $\beta \in \{0, 1, 5, 10\}$, where $\beta = 0$ removes gating entirely (i.e., uniform distillation). The optimal performance is achieved at $\beta = 5$, which effectively balances two distinct failure modes: an excessively small β (including the

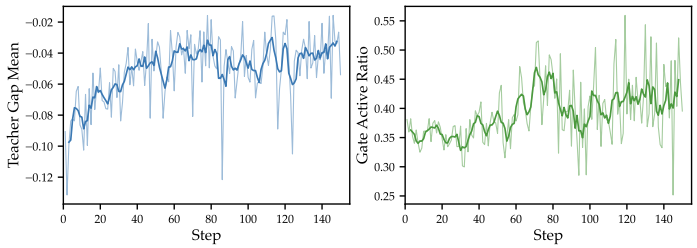


Figure 5: **Training Dynamics.** Average teacher-student gap (Left) and gate activation ratio (Right) during the training of 7B model on ALFWorld.

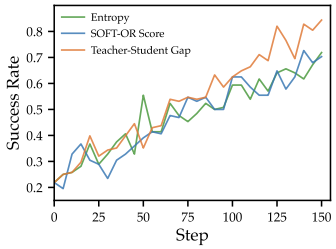


Figure 6: Ablations of Token-level Gating on Qwen2.5-3B-Instruct.

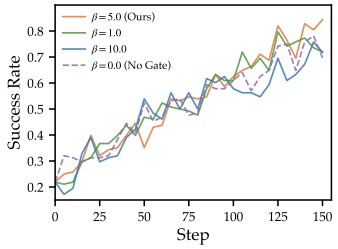


Figure 7: Ablations of β on Qwen2.5-3B-Instruct.

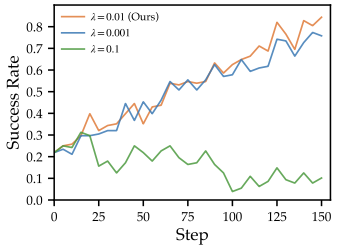


Figure 8: Ablations of λ on Qwen2.5-3B-Instruct.

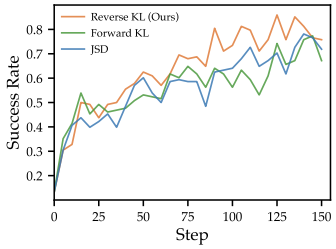


Figure 9: Ablations of $\mathcal{L}_{\text{SDAR}}$ type on Qwen2.5-7B-Instruct.

no-gate baseline) applies distillation indiscriminately, thereby inheriting the multi-turn instability of naïve OPSD; conversely, an overly large β strictly binarizes the gate, stripping away the smooth modulation necessary for assigning partial credit on borderline tokens.

Distillation Coefficient λ. Figure 8 sweeps $\lambda_{\text{SDAR}} \in \{0.001, 0.01, 0.1\}$, identifying $\lambda = 0.01$ as optimal for providing a steady signal without disrupting the primary RL objective. At $\lambda = 0.1$, the distillation gradient dominates, forcing the student toward inferior behaviors due to the teacher's lower confidence in multi-turn settings (Figure 5). Conversely, $\lambda = 0.001$ offers insufficient corrective pressure. These results confirm the necessity of a carefully calibrated, moderate coefficient.

Distillation Objective. Figure 9 compares three token-level objectives on Qwen2.5-7B: reverse KL (default), forward KL, and Jensen–Shannon divergence (JSD), where JSD is defined as:

$$D_{\text{JSD}}^{(t)} = \tfrac{1}{2}\, D_{\text{KL}}\big(\pi_\theta(\cdot \mid s_t) \,\|\, M_t\big) + \tfrac{1}{2}\, D_{\text{KL}}\big(\pi_T(\cdot \mid s_t^+) \,\|\, M_t\big)$$

Reverse KL clearly outperforms the alternatives, consistent with our rationale in Section 3.2. Its *mode-seeking* behavior (Murphy, 2012) lets the student concentrate on modes the teacher supports without having to cover the rest of the teacher's distribution. This selectivity suits our weak, skill-conditioned teacher and complements the gating mechanism. The *mode-covering* forward KL and JSD, by contrast, push the student to spread mass over unreliable guidance, which degrades performance.

5 Conclusion

We presented SDAR, which integrates on-policy self-distillation into RL for multi-turn LLM agents. To counter multi-turn instability and the unreliable negative signals of a skill-conditioned teacher, SDAR keeps RL as the backbone and applies OPSD through a token-level sigmoid gate that amplifies positive teacher endorsements and softly attenuates negative rejections. Across three benchmarks and two model families, it consistently outperforms pure RL and hybrid baselines and remains robust to skill retrieval quality.

AI USE STATEMENT

We used AI assistants in a limited supporting role for language polishing, formatting, and checking submission materials. All AI-assisted outputs were reviewed and verified by the authors before being incorporated. The authors made all decisions regarding the research methodology, experiments, analysis, and presentation, and take full responsibility for the content and results of this work.

ETHICS STATEMENT

This work uses publicly available agent benchmarks (ALFWorld, WebShop, and search-based QA datasets), open-weight Qwen models, and a publicly released skill library, and involves no human participants or sensitive personal data. All environments are simulated or sandboxed: WebShop is a simulated storefront, and search-based QA queries a local retriever rather than the live web, so training and evaluation take no real-world actions. Privileged skills are used only as training-time context and are absent at inference, and all results are reported on standard evaluation splits, with Search-QA separated into in-domain and out-of-domain datasets. Deploying agents trained with SDAR in real environments, where actions can have irreversible consequences, would require additional safety validation, permission controls, and human oversight.

REPRODUCIBILITY STATEMENT

Section 3.2 and Section 3.3 define the training objective and the token-level gate of SDAR, and Algorithm 1 gives the complete training procedure, with the algorithms of all baselines in Appendix B. The experimental setup specifies the backbones, training steps, compute budget, data splits, retriever, and skill library, and Table 4 lists the hyperparameters of SDAR and every baseline. Figures 15–17 provide the exact prompt templates for all three environments, and Appendix A contains the theoretical analysis. Our code is available at `https://anonymous.4open.science/r/SDAR-2B07/`.

REFERENCES

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

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating, 2019. URL https://arxiv.org/abs/1912.12294.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Ken Ding. Hdpo: Hybrid distillation policy optimization via privileged self-distillation, 2026. URL https://arxiv.org/abs/2603.23871.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao,

Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: On-policy distillation of large language models, 2026. URL https://arxiv.org/abs/2306.08543.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision, 2026. URL https://arxiv.org/abs/2604.12002.

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.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

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.

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, 2026. URL https://arxiv.org/abs/2604.02268.

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.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. 2012.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models, 2026. URL https://arxiv.org/abs/2602.04942.

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.

Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. URL https://arxiv.org/abs/1011.0686.

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.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

Zhengliang Shi, Shen Gao, Lingyong Yan, Yue Feng, Xiuyi Chen, Zhumin Chen, Dawei Yin, Suzan Verberne, and Zhaochun Ren. Tool learning in the wild: Empowering language models as automatic tool agents. In *Proceedings of the ACM on Web Conference 2025*, pp. 2222–2237, 2025.

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.

Core Team, Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, Gang Xie, Hailin Zhang, Hanglong Lv, Hanyu Li, Heyu Chen, Hongshen Xu, Houbin Zhang, Huaqiu Liu, Jiangshan Duo, Jianyu Wei, Jiebao Xiao, Jinhao Dong, Jun Shi, Junhao Hu, Kainan Bao, Kang Zhou, Lei Li, Liang Zhao, Linghao Zhang, Peidian Li, Qianli Chen, Shaohui Liu, Shihua Yu, Shijie Cao, Shimao Chen, Shouqiu Yu, Shuo Liu, Tianling Zhou, Weijiang Su, Weikun Wang, Wenhan Ma, Xiangwei Deng, Bohan Mao, Bowen Ye, Can Cai, Chenghua Wang, Chengxuan Zhu, Chong Ma, Chun Chen, Chunan Li, Dawei Zhu, Deshan Xiao, Dong Zhang, Duo Zhang, Fangyue Liu, Feiyu Yang, Fengyuan Shi, Guoan Wang, Hao Tian, Hao Wu, Heng Qu, Hongfei Yi, Hongxu An, Hongyi Guan, Xing Zhang, Yifan Song, Yihan Yan, Yihao Zhao, Yingchun Lai, Yizhao Gao, Yu Cheng, Yuanyuan Tian, Yudong Wang, Zhen Tang, Zhengju Tang, Zhengtao Wen, Zhichao Song, Zhixian Zheng, Zihan Jiang, Jian Wen, Jiarui Sun, Jiawei Li, Jinlong Xue, Jun Xia, Kai Fang, Menghang Zhu, Nuo Chen, Qian Tu, Qihao Zhang, Qiying Wang, Rang Li, Rui Ma, Shaolei Zhang, Shengfan Wang, Shicheng Li, Shuhao Gu, Shuhuai Ren, Sirui Deng, Tao Guo, Tianyang Lu, Weiji Zhuang, Weikang Zhang, Weimin Xiong, Wenshan Huang, Wenyu Yang, Xin Zhang, Xing Yong, Xu Wang, Xueyang Xie, Yilin Jiang, Yixin Yang, Yongzhe He, Yu Tu, Yuanliang Dong, Yuchen Liu, Yue Ma, Yue Yu, Yuxing Xiang, Zhaojun Huang, Zhenru Lin, Zhipeng Xu, Zhiyang Chen, Zhonghua Deng, Zihan Zhang, and Zihao Yue. Mimo-v2-flash technical report, 2026a. URL https://arxiv.org/abs/2601.02780.

Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Meituan LongCat Team, Anchun Gui, Bei Li, Bingyang Tao, Bole Zhou, Borun Chen, Chao Zhang, Chen Gao, Chen Zhang, Chengcheng Han, et al. Longcat-flash-thinking-2601 technical report. *arXiv preprint arXiv:2601.16725*, 2026b.

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

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

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

Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. Tcod: Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents, 2026b. URL https://arxiv.org/abs/2604.24005.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation, 2023. URL https://arxiv.org/abs/2307.15190.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning, 2026. URL https://arxiv.org/abs/2602.08234.

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

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, 2026a. URL https://arxiv.org/abs/2604.03128.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation, 2026b. URL https://arxiv.org/abs/2602.12125.

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.

Jiabo Ye, Xi Zhang, Haiyang Xu, Haowei Liu, Junyang Wang, Zhaoqing Zhu, Ziwei Zheng, Feiyu Gao, Junjie Cao, Zhengxi Lu, et al. Mobile-agent-v3: Fundamental agents for gui automation. *arXiv preprint arXiv:2508.15144*, 2025.

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

Ruixiang Zhang, Richard He Bai, Huangjie Zheng, Navdeep Jaitly, Ronan Collobert, and Yizhe Zhang. Embarrassingly simple self-distillation improves code generation, 2026. URL https://arxiv.org/abs/2604.01193.

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, 2026. URL https://arxiv.org/abs/2601.18734.

A Theoretical Analysis

A.1 Design Rationale

The central design question is how the divergence signal should enter optimization. We adopt the reverse-KL-aligned gap

$$\Delta_t = \log \pi_T(y_t \mid s_t^+) - \log \pi_\theta(y_t \mid s_t)$$

rather than forward KL, because it naturally evaluates on student-sampled tokens and avoids the computationally expensive full-vocabulary matching. However, using this raw gap directly as a coefficient would create overly strong, unbounded token-level gradients during early training or under severe teacher-student mismatch. To resolve this, we wrap the gap in a sigmoid function

$$g_t = \sigma(\beta \Delta_t),$$

which transforms the raw discrepancy into a bounded and monotone importance weight

$$g_t \in (0, 1), \qquad \frac{\partial g_t}{\partial \Delta_t} > 0.$$

This preserves the ordering of token importance while strictly preventing gradient explosion. Finally, we apply a stop-gradient operator to the gate. Detaching g_t ensures it acts purely as a confidence weight rather than creating an additional, self-referential optimization pathway, yielding a stable, first-order weighted likelihood update.

A.2 Theoretical Properties

We formalize the stability and curriculum properties of SDAR through the following propositions.

Proposition 1 (Equivalent Weighted Likelihood Form)**.** *Assume that both* $\log \pi_T(y_t \mid s_t^+)$ *and* g_t *are detached from gradient computation. Minimizing* $\mathcal{L}_{\text{SDAR}}$ *is equivalent, up to an additive constant, to maximizing a token-weighted log-likelihood objective on student-sampled tokens:*

$$\mathcal{L}_{\text{SDAR}} = C - \operatorname{Agg}(g_t \log \pi_\theta(y_t \mid s_t)),$$

where

$$C = \operatorname{Agg}\big(g_t \log \pi_T(y_t \mid s_t^+)\big)$$

is constant with respect to θ*.*

Proof. By definition,

$$\begin{aligned}\mathcal{L}_{\text{SDAR}} &= \operatorname{Agg}\big(g_t\big(\log \pi_T(y_t \mid s_t^+) - \log \pi_\theta(y_t \mid s_t)\big)\big) \\ &= \operatorname{Agg}\big(g_t \log \pi_T(y_t \mid s_t^+)\big) - \operatorname{Agg}(g_t \log \pi_\theta(y_t \mid s_t)) \\ &= C - \operatorname{Agg}(g_t \log \pi_\theta(y_t \mid s_t)),\end{aligned}$$

where the first term $C = \operatorname{Agg}\big(g_t \log \pi_T(y_t \mid s_t^+)\big)$ is constant w.r.t. θ since both g_t and $\log \pi_T(y_t \mid s_t^+)$ are detached. □

Proposition 2 (Gradient Form)**.** *Under the same assumptions, the gradient of* $\mathcal{L}_{\text{SDAR}}$ *is strictly modulated by the bounded scalar gate:*

$$\nabla_\theta \mathcal{L}_{\text{SDAR}} = -\operatorname{Agg}(g_t \nabla_\theta \log \pi_\theta(y_t \mid s_t)).$$

Proof. From Proposition 1,

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\text{SDAR}} &= \nabla_\theta \left[C - \operatorname{Agg}(g_t \log \pi_\theta(y_t \mid s_t))\right] \\ &= 0 - \operatorname{Agg}(g_t \nabla_\theta \log \pi_\theta(y_t \mid s_t)) \\ &= -\operatorname{Agg}(g_t \nabla_\theta \log \pi_\theta(y_t \mid s_t)).\end{aligned}$$

□

Proposition 3 (Monotonicity and Smoothness of the Gate). *The gate $g_t = \sigma(\beta\Delta_t)$ is strictly increasing in Δ_t, inducing an online token-level curriculum where larger discrepancies receive stronger weights. Its derivative satisfies*

$$\frac{\partial g_t}{\partial \Delta_t} = \beta\,\sigma(\beta\Delta_t)\big(1-\sigma(\beta\Delta_t)\big) \in (0,\,\beta/4].$$

Proof. By the chain rule,

$$\frac{\partial g_t}{\partial \Delta_t} = \beta\,\sigma'(\beta\Delta_t).$$

Since the logistic sigmoid satisfies

$$\sigma'(z) = \sigma(z)\big(1-\sigma(z)\big) > 0 \qquad \forall\, z \in \mathbb{R},$$

we obtain

$$\frac{\partial g_t}{\partial \Delta_t} = \beta\,\sigma(\beta\Delta_t)\big(1-\sigma(\beta\Delta_t)\big) > 0.$$

Let $u = \sigma(\beta\Delta_t) \in (0,1)$:

$$u(1-u) \le \left(\frac{u+(1-u)}{2}\right)^2 = \frac{1}{4}$$
$$\frac{\partial g_t}{\partial \Delta_t} = \beta\,u(1-u) \le \frac{\beta}{4}.$$

□

Proposition 4 (Bounded Auxiliary Gradient). *Assume that $\|\nabla_\theta \log \pi_\theta(y_t \mid s_t)\| \le B_t$ for each valid token. Then the gate cannot amplify the auxiliary gradient beyond the unweighted likelihood gradient:*

$$\|\nabla_\theta \mathcal{L}_{\text{SDAR}}\| \le \text{Agg}(B_t).$$

Proof. By Proposition 2,

$$\begin{aligned}\|\nabla_\theta \mathcal{L}_{\text{SDAR}}\| &= \|\text{Agg}(g_t \nabla_\theta \log \pi_\theta(y_t \mid s_t))\| \\ &\le \text{Agg}(g_t\,\|\nabla_\theta \log \pi_\theta(y_t \mid s_t)\|) \\ &\le \text{Agg}(1 \cdot B_t) = \text{Agg}(B_t),\end{aligned}$$

where the first inequality is the triangle inequality and the second uses $0 < g_t < 1$ and $\|\nabla_\theta \log \pi_\theta(y_t \mid s_t)\| \le B_t$. □

Proposition 5 (Effect of Not Detaching the Gate). *Without stop-gradient on the gate, the non-detached token loss $\tilde{\ell}_t = \sigma(\beta\Delta_t)\,\Delta_t$ introduces an unstable self-referential coupling term into the gradient:*

$$\nabla_\theta \tilde{\ell}_t = -\Big(g_t + \beta\Delta_t\, g_t(1-g_t)\Big)\nabla_\theta \log \pi_\theta(y_t \mid s_t).$$

Proof. Write $\tilde{\ell}_t = g_t\Delta_t$. Since $\log \pi_T(y_t \mid s_t^+)$ is constant w.r.t. θ,

$$\begin{aligned}\nabla_\theta \Delta_t &= \nabla_\theta\big[\log \pi_T(y_t \mid s_t^+) - \log \pi_\theta(y_t \mid s_t)\big] \\ &= -\nabla_\theta \log \pi_\theta(y_t \mid s_t).\end{aligned}$$

By the chain rule on $g_t = \sigma(\beta\Delta_t)$,

$$\begin{aligned}\nabla_\theta g_t &= \beta\,\sigma'(\beta\Delta_t)\,\nabla_\theta\Delta_t = \beta\,g_t(1-g_t)\,\nabla_\theta\Delta_t \\ &= -\beta\,g_t(1-g_t)\,\nabla_\theta \log \pi_\theta(y_t \mid s_t).\end{aligned}$$

Applying the product rule,

$$\begin{aligned}\nabla_\theta \tilde{\ell}_t &= (\nabla_\theta g_t)\,\Delta_t + g_t\,(\nabla_\theta \Delta_t) \\ &= \big[-\beta\,g_t(1-g_t)\,\nabla_\theta \log \pi_\theta(y_t \mid s_t)\big]\,\Delta_t \\ &\quad + g_t\,\big[-\nabla_\theta \log \pi_\theta(y_t \mid s_t)\big] \\ &= -\beta\,g_t(1-g_t)\,\Delta_t\,\nabla_\theta \log \pi_\theta(y_t \mid s_t) \\ &\quad - g_t\,\nabla_\theta \log \pi_\theta(y_t \mid s_t) \\ &= -\Big(g_t + \beta\Delta_t\,g_t(1-g_t)\Big)\,\nabla_\theta \log \pi_\theta(y_t \mid s_t).\end{aligned}$$

□

B ALGORITHM

The full procedure of SDAR is presented in Algorithm 1. We compare against five baselines listed below:

- **GRPO** (Shao et al., 2024) (Algorithm 2): RL baseline that optimizes the policy via a clipped surrogate objective with group-relative advantages.
- **OPSD** (Zhao et al., 2026) (Algorithm 3): an on-policy self-distillation method that distills token-level knowledge from a frozen reference policy π_{ref} into the student.
- **Skill-SD** (Wang et al., 2026a) (Algorithm 4): a hybrid method that augments GRPO with an importance-weighted K_3-divergence distillation loss, using retrieved skills as privileged context to construct the teacher signal.
- **GRPO+OPSD** (Algorithm 5): a hybrid method that simply adds the OPSD distillation loss from π_{ref} as an auxiliary objective on top of GRPO training.
- **RLSD** (Yang et al., 2026a) (Algorithm 6): a hybrid method that re-weights GRPO's advantages with self-teacher's gap.

Algorithm 1 SDAR

Require: Policy π_θ, task set $\mathcal{S}$, skill library $\mathcal{E} = \{e_1, \ldots, e_M\}$, group size G, mixing coefficient λ, sharpness β, clip bound ϵ

for each training iteration **do**
  Sample a batch of tasks $\{x\}$ from $\mathcal{S}$
  for each task x **do**
    Retrieve skill c^+ from $\mathcal{E}$ ▷ UCB / KM / Full / Random
    // Step 1: On-policy rollout
    Sample G responses $\{y^{(1)}, \ldots, y^{(G)}\} \sim \pi_\theta(\cdot \mid x)$
    // Step 2: Sequence-level advantage from environment
    for $i = 1, \ldots, G$ **do**
      Obtain reward $R(x, y^{(i)})$ from environment interaction
    end for
    Compute $A^{(i)} = \frac{R(x,y^{(i)}) - \mu_G}{\sigma_G}$ ▷ Group-relative advantage
    // Step 3: GRPO policy loss
    for $i = 1, \ldots, G$ **do**
      for $t = 1, \ldots, |y^{(i)}|$ **do**
        $r_t^{(i)} \leftarrow \pi_\theta(y_t^{(i)} \mid s_t^{(i)}) \,/\, \pi_{\theta_{\text{old}}}(y_t^{(i)} \mid s_t^{(i)})$
      end for
    end for
    Compute $\mathcal{L}_{\text{GRPO}}$ via clipped surrogate with $\{A^{(i)}, r_t^{(i)}\}$
    // Step 4: Token-level gated distillation
    for $i = 1, \ldots, G$ **do**
      Compute teacher logits via forward pass with $(x, c^+, y^{(i)})$
      for $t = 1, \ldots, |y^{(i)}|$ **do**
        $\Delta_t \leftarrow \text{sg}\big(\log \pi_\theta(y_t^{(i)} \mid s_t^+) - \log \pi_\theta(y_t^{(i)} \mid s_t)\big)$
        $g_t \leftarrow \sigma(\beta \cdot \Delta_t)$
        $\ell_t \leftarrow g_t \cdot \big(\log \pi_\theta(y_t^{(i)} \mid s_t^+) - \log \pi_\theta(y_t^{(i)} \mid s_t)\big)$
      end for
    end for
    $\mathcal{L}_{\text{SDAR}} \leftarrow \frac{1}{G} \sum_{i=1}^{G} \text{Agg}\big(\ell_t^{(i)}\big)$
    // Step 5: Joint policy update
    Update θ by minimizing $\mathcal{L}(\theta) = \mathcal{L}_{\text{GRPO}}(\theta) + \lambda \cdot \mathcal{L}_{\text{SDAR}}(\theta)$
  end for
end for

Algorithm 2 GRPO

Require: Policy π_θ, task set $\mathcal{S}$, group size G, clip bounds $\epsilon_{\text{lo}}, \epsilon_{\text{hi}}$, dual-clip constant c

for each training iteration **do**
  Sample a batch of tasks $\{x\}$ from $\mathcal{S}$
  for each task x **do**
    // Step 1: On-policy rollout
    Sample G responses $\{y^{(1)}, \ldots, y^{(G)}\} \sim \pi_\theta(\cdot \mid x)$
    // Step 2: Sequence-level advantage from environment
    for $i = 1, \ldots, G$ **do**
      Obtain reward $R(x, y^{(i)})$ from environment interaction
    end for
    Compute $A^{(i)} = \frac{R(x,y^{(i)}) - \mu_G}{\sigma_G}$ ▷ Group-relative advantage
    // Step 3: Clipped surrogate policy loss
    for $i = 1, \ldots, G$ **do**
      for $t = 1, \ldots, |y^{(i)}|$ **do**
        $r_t^{(i)} \leftarrow \pi_\theta(y_t^{(i)} \mid s_t^{(i)}) \,/\, \pi_{\theta_{\text{old}}}(y_t^{(i)} \mid s_t^{(i)})$
        $L_1 \leftarrow -A^{(i)} r_t^{(i)}$
        $L_2 \leftarrow -A^{(i)} \,\text{clip}(r_t^{(i)},\, 1{-}\epsilon_{\text{lo}},\, 1{+}\epsilon_{\text{hi}})$
        $\ell_t^{(i)} \leftarrow \begin{cases} \min(-A^{(i)} c,\ \max(L_1, L_2)) & \text{if } A^{(i)} < 0 \\ \max(L_1, L_2) & \text{otherwise} \end{cases}$
      end for
    end for
    $\mathcal{L}_{\text{GRPO}} \leftarrow \text{Agg}(\{\ell_t^{(i)}\})$
    // Step 4: Policy update
    Update θ by minimizing $\mathcal{L}(\theta) = \mathcal{L}_{\text{GRPO}}(\theta)$
  end for
end for

Algorithm 3 OPSD

Require: Policy π_θ, frozen reference π_{ref}, task set $\mathcal{S}$, group size G, KL coefficient α

for each training iteration **do**
  Sample a batch of tasks $\{x\}$ from $\mathcal{S}$
  for each task x **do**
    // Step 1: On-policy rollout
    Sample G responses $\{y^{(1)}, \ldots, y^{(G)}\} \sim \pi_\theta(\cdot \mid x)$
    // Step 2: Token-level KL distillation from reference
    for $i = 1, \ldots, G$ **do**
      for $t = 1, \ldots, |y^{(i)}|$ **do**
        $d_t^{(i)} \leftarrow \log \pi_\theta(y_t^{(i)} \mid s_t) - \log \pi_{\text{ref}}(y_t^{(i)} \mid s_t)$ ▷ $D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$
      end for
    end for
    $\mathcal{L}_{\text{OPSD}} \leftarrow \alpha \cdot \text{Agg}(\{d_t^{(i)}\})$
    // Step 3: Policy update
    Update θ by minimizing $\mathcal{L}(\theta) = \mathcal{L}_{\text{OPSD}}(\theta)$
  end for
end for

C HYPERPARAMETERS

Table 4 summarizes the method-specific hyperparameters used for all baselines and SDAR across our experiments.

Algorithm 4 Skill-SD

Require: Policy π_θ, task set $\mathcal{S}$, skill library $\mathcal{E} = \{e_1, \ldots, e_M\}$, group size G, distillation coefficient λ, clip bound ϵ

for each training iteration **do**
  Sample a batch of tasks $\{x\}$ from $\mathcal{S}$
  for each task x **do**
    Retrieve skill c^+ from $\mathcal{E}$ ▷ UCB
    // Step 1: On-policy rollout
    Sample G responses $\{y^{(1)}, \ldots, y^{(G)}\} \sim \pi_\theta(\cdot \mid x)$
    // Step 2: Sequence-level advantage from environment
    for $i = 1, \ldots, G$ **do**
      Obtain reward $R(x, y^{(i)})$ from environment interaction
    end for
    Compute $A^{(i)} = \frac{R(x,y^{(i)}) - \mu_G}{\sigma_G}$ ▷ Group-relative advantage
    // Step 3: GRPO policy loss (same as Algorithm 2)
    Compute $\mathcal{L}_{\text{GRPO}}$ via clipped surrogate with $\{A^{(i)}, r_t^{(i)}\}$
    // Step 4: Importance-weighted K3 distillation
    for $i = 1, \ldots, G$ **do**
      Compute teacher log-probs via forward pass with $(x, c^+, y^{(i)})$
      for $t = 1, \ldots, |y^{(i)}|$ **do**
        $d_t \leftarrow \log \pi_\theta(y_t^{(i)} \mid s_t) - \log \pi_\theta(y_t^{(i)} \mid s_t^+)$ ▷ Student − Teacher
        $k_t \leftarrow \exp(-d_t) - 1 + d_t$ ▷ K_3 divergence
        $\rho_t \leftarrow \exp\big(\log \pi_\theta(y_t^{(i)} \mid s_t) - \log \pi_{\theta_{\text{old}}}(y_t^{(i)} \mid s_t)\big)$ ▷ On-policy IS ratio
        $\ell_t^{(i)} \leftarrow \rho_t \cdot k_t$
      end for
    end for
    $\mathcal{L}_{\text{Skill-SD}} \leftarrow \text{Agg}\big(\{\ell_t^{(i)}\}\big)$
    // Step 5: Joint policy update
    Update θ by minimizing $\mathcal{L}(\theta) = \mathcal{L}_{\text{GRPO}}(\theta) + \lambda \cdot \mathcal{L}_{\text{Skill-SD}}(\theta)$
  end for
end for

Table 4: **Hyperparameters**. η: learning rate; G: group size; ϵ: PPO clip ratio; λ: distillation loss coefficient; β: sigmoid gate sharpness; α_{KL}: KL penalty coefficient toward the reference policy; SRS: skill retrieval strategy (KM = keyword matching).

Method	η	G	ϵ	λ	β	α_{KL}	SRS
GRPO	10^{-6}	8	0.2	—	—	0.01	—
Skill-GRPO	10^{-6}	8	0.2	—	—	0.01	KM
OPSD	10^{-6}	—	—	0.01	5.0	0.01	KM
Skill-SD	10^{-6}	8	0.2	0.001	—	0.01	KM
GRPO+OPSD	10^{-6}	8	0.2	0.01	0.0	0.01	KM
RLSD	10^{-6}	8	0.2	0.5	—	0.01	KM
SDAR (Ours)	10^{-6}	8	0.2	0.01	5.0	0.01	KM

D TRAINING DYNAMICS

We present the full training dynamics of SDAR across all model scales and environments in Figures 10–14, tracking five diagnostic metrics throughout training.

Algorithm 5 GRPO+OPSD

Require: Policy π_θ, frozen reference π_{ref}, task set $\mathcal{S}$, group size G, KL coefficient α, clip bound ϵ

for each training iteration **do**
  Sample a batch of tasks $\{x\}$ from $\mathcal{S}$
  for each task x **do**
    // Step 1: On-policy rollout
    Sample G responses $\{y^{(1)}, \ldots, y^{(G)}\} \sim \pi_\theta(\cdot \mid x)$
    // Step 2: Sequence-level advantage from environment
    for $i = 1, \ldots, G$ **do**
      Obtain reward $R(x, y^{(i)})$ from environment interaction
    end for
    Compute $A^{(i)} = \frac{R(x,y^{(i)}) - \mu_G}{\sigma_G}$ ▷ Group-relative advantage
    // Step 3: GRPO policy loss (same as Algorithm 2)
    Compute $\mathcal{L}_{\text{GRPO}}$ via clipped surrogate with $\{A^{(i)}, r_t^{(i)}\}$
    // Step 4: Token-level KL penalty toward π_{ref}
    for $i = 1, \ldots, G$ **do**
      for $t = 1, \ldots, |y^{(i)}|$ **do**
        $d_t^{(i)} \leftarrow \log \pi_\theta(y_t^{(i)} \mid s_t) - \log \pi_{\text{ref}}(y_t^{(i)} \mid s_t)$
      end for
    end for
    $\mathcal{L}_{\text{OPSD}} \leftarrow \alpha \cdot \text{Agg}\big(\{d_t^{(i)}\}\big)$
    // Step 5: Joint policy update
    Update θ by minimizing $\mathcal{L}(\theta) = \mathcal{L}_{\text{GRPO}}(\theta) + \mathcal{L}_{\text{OPSD}}(\theta)$
  end for
end for

E PROMPT

Figures 15–17 present the full prompt templates used by SDAR for the three evaluation environments, where `{skill_context}` is populated with the retrieved skill during training and left empty at inference time.

Algorithm 6 RLSD

Require: Policy π_θ, task set $\mathcal{S}$, skill library $\mathcal{E} = \{e_1, \ldots, e_M\}$, group size G, mixing coefficient λ, weight clip bound ϵ_w, policy clip bound ϵ

for each training iteration **do**
  Sample a batch of tasks $\{x\}$ from $\mathcal{S}$
  for each task x **do**
    Retrieve skill c^+ from $\mathcal{E}$ ▷ UCB / KM / Full / Random
    // Step 1: On-policy rollout
    Sample G responses $\{y^{(1)}, \ldots, y^{(G)}\} \sim \pi_\theta(\cdot \mid x)$
    // Step 2: Sequence-level advantage from environment
    for $i = 1, \ldots, G$ **do**
      Obtain reward $R(x, y^{(i)})$ from environment interaction
    end for
    Compute $A^{(i)} = \frac{R(x,y^{(i)}) - \mu_G}{\sigma_G}$ ▷ Group-relative advantage
    // Step 3: Token-level advantage reweighting via teacher
    for $i = 1, \ldots, G$ **do**
      Compute teacher log-probs via forward pass with $(x, c^+, y^{(i)})$
      for $t = 1, \ldots, |y^{(i)}|$ **do**
        $\delta_t \leftarrow \log \pi_\theta(y_t^{(i)} \mid s_t^+) - \log \pi_{\theta_{\text{old}}}(y_t^{(i)} \mid s_t)$ ▷ Teacher − Student gap
        $w_t \leftarrow \text{clip}\big(\exp(\text{sign}(A^{(i)}) \cdot \delta_t),\ 1-\epsilon_w,\ 1+\epsilon_w\big)$
        $\hat{A}_t^{(i)} \leftarrow A^{(i)} \cdot \big[(1-\lambda) + \lambda \cdot w_t\big]$
      end for
    end for
    // Step 4: Clipped surrogate with token-level advantages
    Compute $\mathcal{L}_{\text{RLSD}}$ via clipped surrogate with $\{\hat{A}_t^{(i)},\ r_t^{(i)}\}$
    // Step 5: Policy update
    Update θ by minimizing $\mathcal{L}(\theta) = \mathcal{L}_{\text{RLSD}}(\theta)$
  end for
end for

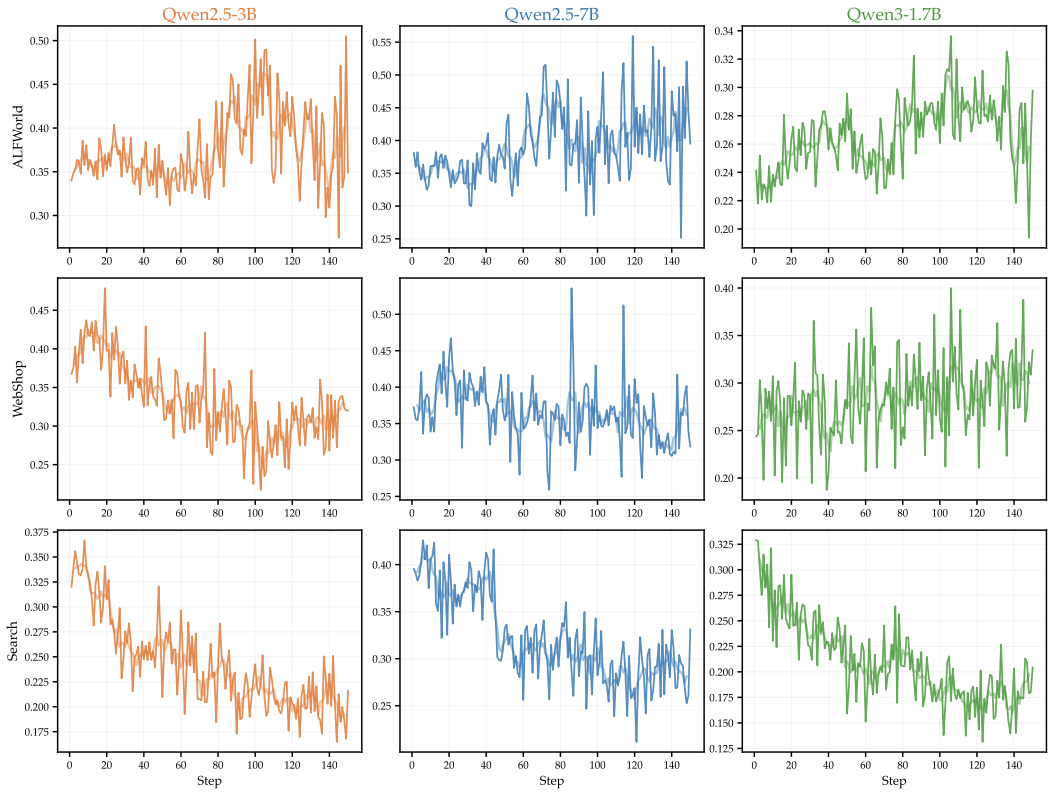


Figure 10: **Gate Active Ratio** when training Qwen2.5-3B, Qwen2.5-7B and Qwen3-1.7B on ALFWorld, WebShop and Search-QA.

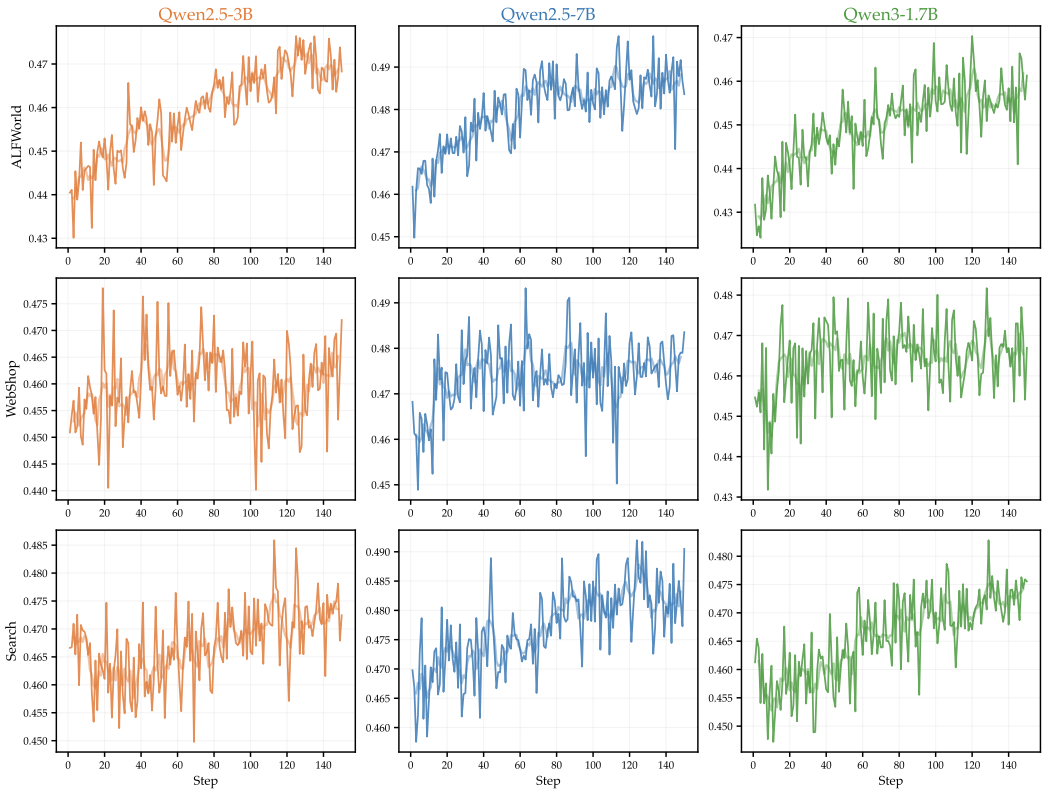


Figure 11: **Gate Mean** when training Qwen2.5-3B, Qwen2.5-7B and Qwen3-1.7B on ALFWorld, WebShop and Search-QA.

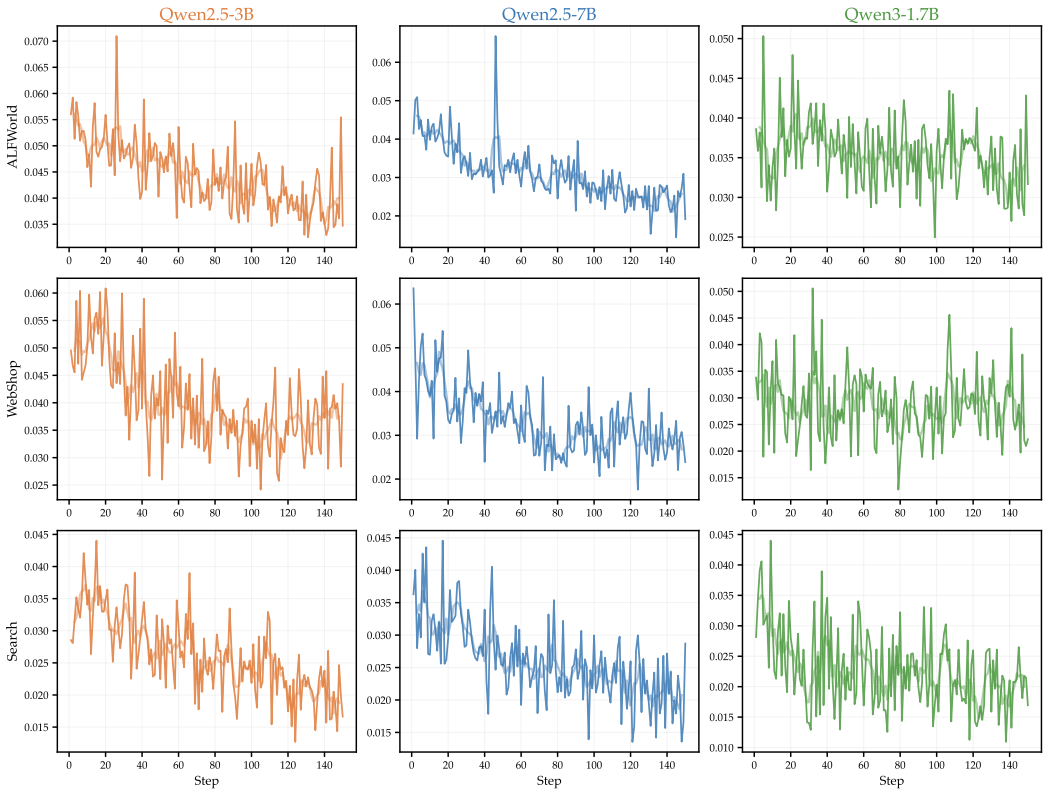


Figure 12: **OPSD Loss** when training Qwen2.5-3B, Qwen2.5-7B and Qwen3-1.7B on ALFWorld, WebShop and Search-QA.

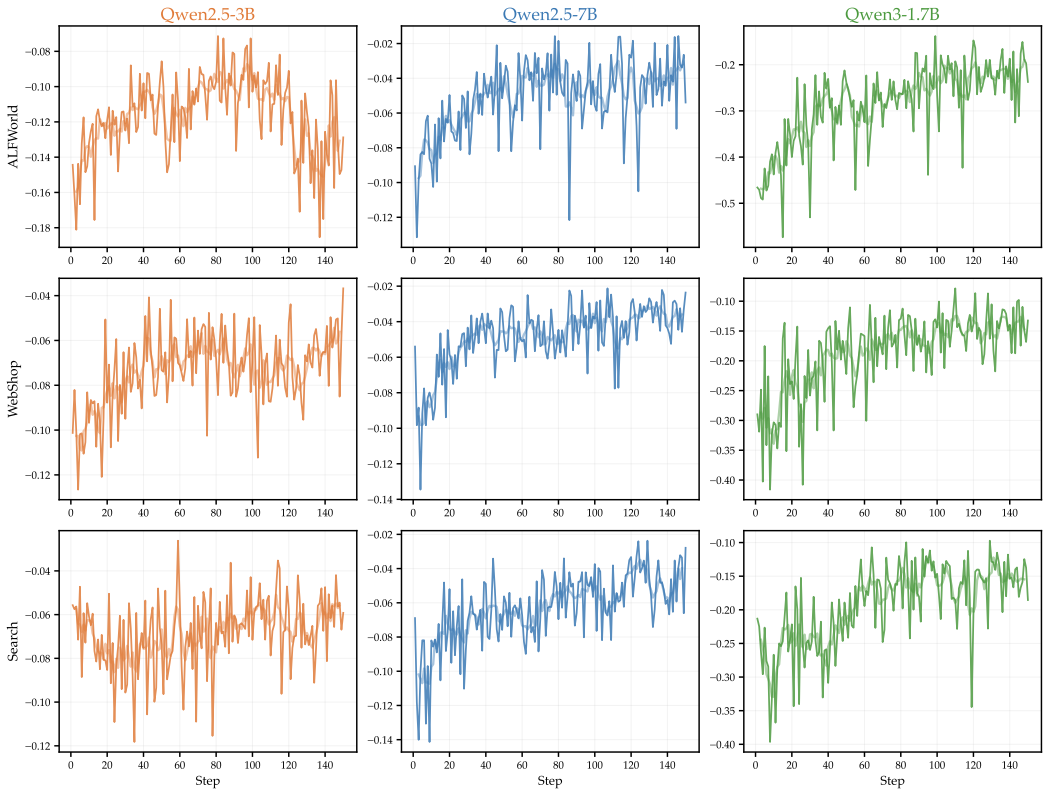


Figure 13: **Teacher-Student Gap** when training Qwen2.5-3B, Qwen2.5-7B and Qwen3-1.7B on ALFWorld, WebShop and Search-QA.

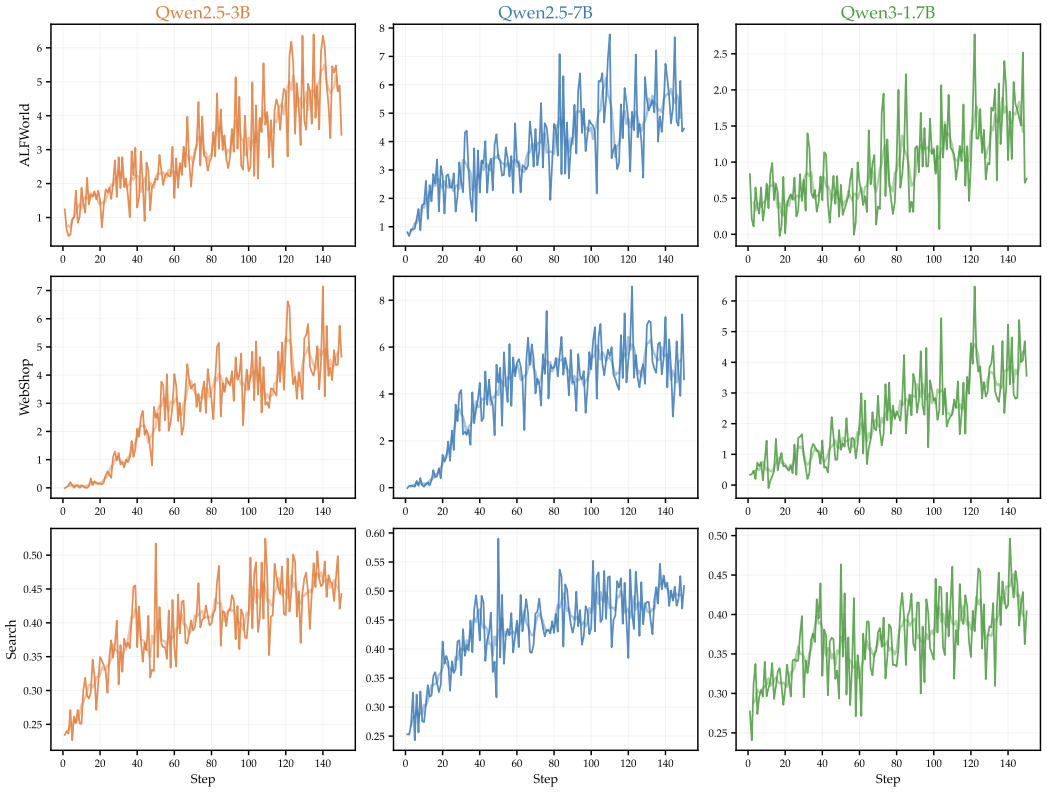


Figure 14: **Reward Curve** when training Qwen2.5-3B, Qwen2.5-7B and Qwen3-1.7B on ALFWorld, WebShop and Search-QA.

Prompt of SDAR on ALFWorld

You are an expert agent operating in the ALFRED Embodied Environment. Your task is to: {task_description}.
{skill_context}
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.

Figure 15: Prompt template used by SDAR for the ALFWorld task environment.

Prompt of SDAR on Search-based QA

You are an expert agent tasked with answering the given question step-by-step.
{skill_context}
Your question: {task_description}.
Prior to this step, you have already taken {step_count} step(s). Below is the interaction history where <search> </search> wrapped your past search queries and <information> </information> wrapped the corresponding search results returned by the external search engine. History:
{memory_context}
Now it's your turn to respond for the current step. You should first conduct a reasoning process. This process MUST be enclosed within <think> </think> tags. After completing your reasoning, choose only one of the following actions (do not perform both):

1. If you find you lack some knowledge, you **MUST** call a search engine to get more external information using format: <search> your query </search>.
2. If you have enough knowledge to answer the question confidently, provide your final answer within <answer> </answer> tags, without detailed illustrations. For example, <answer>Beijing</answer>.

Figure 16: Prompt template used by SDAR for the Search-based QA task environment.

Prompt of SDAR on WebShop

You are an expert autonomous agent operating in the WebShop e-commerce environment.
{skill_context}
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.

Figure 17: Prompt template used by SDAR for the WebShop task environment.