STATE TRANSITION MATTERS: SELF-DISTILLED CREDIT ASSIGNMENT FOR MULTI-TURN AGENTIC RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has improved large language models (LLMs) on verifiable tasks, but training multi-turn interactive agents remains challenging due to sparse outcome rewards and coarse credit assignment. Existing methods mainly rely on outcome reward back-propagation to assign credit, ignoring the rich task-relevant information hidden in state transitions, which naturally reveal the local consequences of actions. Based on this observation, we propose **SDIAL**, a novel **S**elf-**Di**stilled **A**dvantage reshaping method for multi-turn credit assignment in reinforcement **L**earning. In SDIAL, the LLM of student mode interacts with the environment, while the same LLM of teacher mode leverages the state transitions to provide hindsight evaluation of the student's actions. The difference between the teacher's and student's output distributions is distilled into token-level credit signals to refine advantage estimates at each decision step. To maintain the effectiveness of self-distillation and stabilize policy optimization, SDIAL introduces an asymmetric reshaping mechanism that applies different operators to reshape positive and negative advantages. Experiments on multiple long-horizon interactive benchmarks show that SDIAL improves training stability and robustness while achieving strong task performance across diverse environments. Code is available at `https://anonymous.4open.science/r/SDIAL`.

1 INTRODUCTION

Recent advances in reinforcement learning (RL) (Cao et al., 2024) have shown strong performance on single-turn, verifiable tasks such as mathematical reasoning (Ahn et al., 2024; Yan et al., 2025). However, extending these successes to multi-turn interactive agent settings remains challenging. Unlike single-turn tasks, multi-turn tasks must make sequential decisions over long horizons while receiving only sparse outcome rewards from the environment, with limited step-level supervision (Zeng et al., 2025). While classical RL can perform credit assignment using only the outcome rewards (Seo et al., 2019), they perform poorly in agentic RL with LLMs due to the large action space, non-Markovian environments and deactivation of value functions (Kazemnejad et al., 2024).

The credit assignment in multi-turn agentic RL faces the following fundamental challenges. First, the reward signal is inherently sparse in multi-turn agentic RL, where the environment typically provides only a final outcome reward indicating task success or failure, rather than process-level rewards for intermediate decisions (Wei et al., 2025). Second, each agentic decision is generated as a long sequence of tokens, making token-level credit assignment much more complex than action-level credit assignment in classical RL, especially across multiple turns. Moreover, multi-turn interactive environments are often non-markovian and uncertain, which requires the credit assignment mechanism robust enough to unpredictable environmental rewards and transitions.

To improve credit assignment in multi-turn agentic RL, prior works have introduced process reward models (PRMs) (Xi et al., 2025), which provide dense step-level rewards for intermediate decisions. While this alleviates the sparsity of reward signals, it may bring significant inductive bias as the PRMs often rely on extra assumptions. More recent group-based methods, such as GiGPO (Feng et al., 2025), improve credit assignment by introducing two-stage, fine-grained advantage computations. However, these methods still largely rely on the final outcome rewards, making them vulnerable to noisy or incomplete feedbacks. Overall, although existing works have started to explore fine-grained

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.

step-level advantage computation, they ignore the great potential use of state transitions in credit assignment.

To address these limitations, we propose **SDIAL**, a novel **S**elf-**Di**stilled **A**dvantage reshaping method for multi-turn credit assignment in reinforcement **L**earning. During training, the LLM serves as a student policy to collect on-policy trajectories and outcome rewards. As in GRPO, these outcome rewards are normalized within each rollout group to compute original advantages, which reflect overall task performance but assign identical credit to all actions in the same trajectory. Our key observation is that state transitions provide step-level hindsight evidence beyond sparse rewards, as the next state reveals the temporal consequence of each executed action. Because such textual feedback cannot be directly used as supervision, SDIAL reuses the same LLM as a teacher policy conditioned on transition-based hindsight information. The resulting teacher-student distributional discrepancy is distilled into token-level self-distilled action quality scores, which reshape the original advantages. SDIAL further introduces an asymmetric reshaping mechanism based on the sign of the original advantage. Positive advantages usually indicate reward-supported actions, so SDIAL applies a multiplicative operator to adjust reinforcement strength while preserving the update direction. In contrast, negative advantages do not necessarily indicate that the corresponding actions are suboptimal in multi-turn interactions, so SDIAL adopts a more conservative additive operator to adjust their credit. Finally, SDIAL calibrates the reshaping strength with teacher entropy to down-weight uncertain teacher signals. Our main contributions are summarized as follows.

- We propose SDIAL, a self-distilled credit assignment framework that uses state transitions as hindsight feedback to reshape original advantages, enabling token-level credit beyond coarse trajectory-level rewards.
- We develop an entropy-aware confidence calibration mechanism that adaptively controls the strength of self-distilled credit signals, enabling bounded residual refinement of reward-derived advantages and more stable policy optimization.
- We evaluate SDIAL across diverse long-horizon interactive environments, from structured embodied reasoning to open-ended web-based decision-making, showing consistent gains in task performance, training stability, and credit-assignment behavior.

2 Related Works

Reinforcement learning with verifiable rewards. Reinforcement learning with verifiable rewards (RLVR) (Lambert et al., 2024; Su et al., 2025) has become an effective paradigm for improving the reasoning ability of large language models (LLMs). Instead of relying on human preference labels or learned reward models, RLVR optimizes LLM policies with automatically verifiable signals, such as exact-match answers, program execution results, or rule-based checkers (Wen et al., 2025). Representative methods, including PPO-style optimization, RLOO (Ahmadian et al., 2024), and GRPO (Shao et al., 2024), have shown strong performance on single-turn verifiable tasks such as mathematical reasoning (Zhang & Zuo, 2025; Wang et al., 2025) and code generation (Jiang et al., 2025; Zhu et al., 2025). However, these methods are mainly designed for single-turn settings, where each response is evaluated by a final reward and optimized as a whole (Yi et al., 2025). This assumption becomes insufficient for multi-turn agentic RL, where task success depends on a long sequence of intermediate decisions under sparse outcome reward. As a result, directly applying RLVR methods to long-horizon interactive environments often leads to coarse credit assignment, unstable optimization, and inefficient learning (Zeng et al., 2025).

Agentic RL in Multi-Turn Environments. Recent agentic RL methods extend LLM policy optimization to long-horizon interactive environments, where agents make sequences of interdependent decisions before receiving feedback (Li et al., 2026). Policy optimization methods based on final outcome rewards, such as GRPO (Shao et al., 2024) and RLOO (Ahmadian et al., 2024), provide only coarse trajectory-level credit, making it difficult to identify which intermediate decisions should be reinforced or suppressed. Process reward models (PRMs) provide step-level feedback for intermediate reasoning or interaction steps (Xi et al., 2025), yet require additional reward estimators and may suffer from reward-model bias when step-level correctness is hard to verify. Recent group-based methods, such as GiGPO (Feng et al., 2025), improve credit estimation through two-stage advantage computation, first estimating trajectory-level advantages from outcome rewards and then redistributing them across finer-grained groups or decisions. Despite these advances, existing methods still

derive credit primarily from reward-based signals, underusing state-transition feedback that directly reflects action consequences and is crucial for robust learning in dynamic multi-turn environments.

Self-distillation for LLM agents. Self-distillation (Shenfeld et al., 2026; Zhao et al., 2026) improves LLM policy learning by letting the same model serve as both student and teacher under different contexts: the student follows the standard task prompt, while the teacher receives privileged information, such as reference answers, verifier feedback, or richer textual feedback, to provide dense guidance (Shenfeld et al., 2026; Hübotter et al., 2026). When combined with RL, existing methods mainly follow two paradigms. When integrating self-distillation with RL, existing approaches typically follow two paradigms. The first incorporates an explicit distillation objective into the RL loss (Wang et al., 2026a), which may conflict with reward-driven updates when teacher preferences disagree with environmental rewards. The second paradigm uses self-distilled signals only as token-level reweighting factors (Yang et al., 2026), letting rewards determine update directions while self-distilled feedback modulates update magnitudes. Yet, pure reweighting only redistributes existing advantage signals, offering limited correction when the original advantages are weak or noisy. In contrast, SDIAL applies asymmetric advantage reshaping: self-distilled signals multiplicatively reweight positive advantages to avoid interfering with reward-induced updates, while additively correcting negative advantages to provide explicit corrective signals. It also uses naturally available state transitions as hindsight feedback, avoiding extra privileged-information construction.

3 PRELIMINARIES.

3.1 GROUP-BASED POLICY OPTIMIZATION

Group-based Policy Optimization. Group-based policy optimization methods, such as GRPO (Shao et al., 2024) and RLOO (Ahmadian et al., 2024), provide a critic-free framework for LLM reinforcement learning. Given an input prompt or decision context x, they sample a group of trajectories $\{\tau^i\}_{i=1}^G$ and estimate relative advantages by comparing outcome rewards within the group:

$$\hat{A}^i = \frac{R^i - \mu_{\mathcal{G}}}{\sigma_{\mathcal{G}} + \epsilon}, \tag{1}$$

where R^i is the trajectory-level reward, and $\mu_{\mathcal{G}}$ and $\sigma_{\mathcal{G}}$ are the group-level reward statistics.

Different methods vary in group construction, normalization, or advantage redistribution, but they still largely rely on outcome-reward-derived advantages. In multi-turn agentic RL, where rewards are often available only after the full interaction, this assigns the same coarse advantage to all intermediate decisions in a trajectory. SDIAL builds on this group-based framework and reshapes reward-derived advantages with transition-based self-distilled signals for finer-grained credit assignment.

3.2 SELF-DISTILLATION FOR LLMS

Self-distillation provides dense supervision for LLM policy learning without introducing an external teacher model. The same model is instantiated under two contexts: the student is conditioned on the standard decision context, while the teacher additionally receives training-time privileged information, such as reference answers, verifier feedback, environment feedback, or trajectory summaries (Shenfeld et al., 2026; Zhao et al., 2026; Hübotter et al., 2026; Wang et al., 2026a). Given a student-generated trajectory τ^i and its decision prefix h_t^i, the student and teacher distributions can be written as

$$p_{i,t}^S(\cdot) = \pi_\theta(\cdot \mid h_t^i), \qquad p_{i,t}^T(\cdot) = \pi_\theta(\cdot \mid h_t^i, f_t^i), \tag{2}$$

where f_t^i denotes privileged information available only during training. The teacher does not generate a separate trajectory, but re-evaluates the student's trajectory under the enriched context.

Existing methods mainly use the teacher-student discrepancy in two ways. One line treats the teacher as a distillation target and adds a divergence objective to policy optimization:

$$\mathcal{L}_{\mathrm{SD}} = \sum_{i,t} \mathrm{KL}\left(p_{i,t}^S \,\middle\|\, \mathrm{stopgrad}\left[p_{i,t}^T\right]\right). \tag{3}$$

While this provides dense supervision, the distillation objective may conflict with reward-driven updates when teacher preferences disagree with environmental returns.

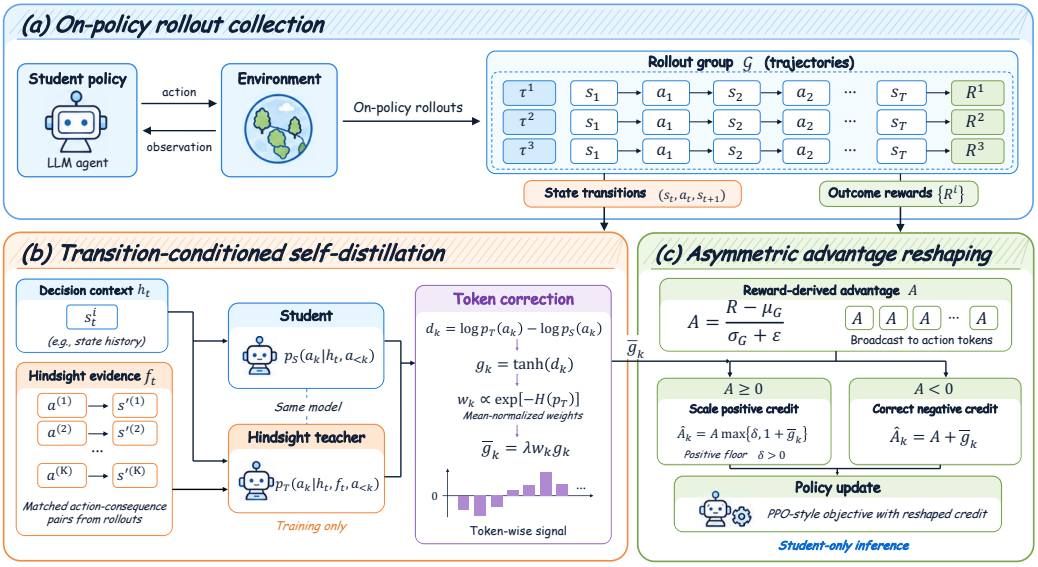


Figure 1: Overview of SDIAL. (a) The student policy collects on-policy multi-turn trajectories. (b) At each step, SDIAL constructs a hindsight teacher and produces an entropy-aware, token-level self-distilled action quality score. (c) This score is used to asymmetrically reshape trajectory-level reward-derived advantages, scaling positive advantages and adding to negative ones to produce fine-grained, token-wise credit.

Another line uses self-distillation as a weighting signal rather than a separate loss. For a sampled action or token a_t^i, the self-distilled signal is commonly derived from the log-probability difference

$$d_t^i = \log p_{i,t}^T(a_t^i) - \log p_{i,t}^S(a_t^i). \tag{4}$$

Methods such as RLSD (Yang et al., 2026) use this signal to modulate token-level update magnitudes while relying on environment rewards to determine update directions. However, using self-distillation only as a reweighting signal may underutilize its corrective value when reward-derived advantages are weak or degenerate. SDIAL builds on this formulation but uses state transitions as privileged hindsight feedback and integrates d_t^i into reward-derived advantage through an entropy-aware asymmetric advantage reshaping mechanism.

4 METHODS

Motivation. SDIAL reshapes the original reward-derived advantage instead of replacing it. Trajectory-level outcome rewards provide the basic update direction, but assign the same advantage to all actions in a trajectory. Although state transitions reveal step-level action consequences, textual transition feedback cannot be directly used as scalar advantages or ground-truth actions. We therefore use self-distillation to convert transition-based hindsight information into token-level action quality scores: the teacher-student discrepancy measures how hindsight changes the model's assessment of the executed action. These scores are then used to refine the original advantage while preserving outcome-grounded optimization.

Overview. Figure 1 illustrates the overall framework of SDIAL. The student policy first collects on-policy multi-turn trajectories with sparse outcome rewards. SDIAL then constructs a hindsight teacher using state transitions from the same or semantically similar decision contexts, and distills the teacher-student discrepancy into token-level *entropy-aware self-distilled action quality score*. During asymmetric advantage reshaping, positive trajectory-level advantages are multiplicatively scaled, while negative advantages are additively adjusted by the gated score, producing fine-grained token-wise credit for each decision step.

4.1 ROLLOUTS COLLECTION

At each training iteration, the student policy interacts with the environment to collect a group of on-policy multi-turn trajectories. Formally, the rollout group is denoted as

$$\mathcal{G} = \{\tau^i\}_{i=1}^{G}, \quad \tau^i = (s_1^i, a_1^i, s_2^i, a_2^i, \ldots, s_T^i, R^i), \tag{5}$$

where G is the rollout group size and R^i denotes the trajectory-level outcome reward.

The collected rollouts serve two purposes in SDIAL. First, their outcome rewards are later used to compute the original reward-derived advantages for policy optimization. Second, their state transitions provide the hindsight information used to construct the self-distillation teacher.

4.2 SELF-DISTILLATION FRAMEWORK FOR ACTION QUALITY SCORE COMPUTATION

Hindsight Teacher Modeling with State Transitions. Unlike single-turn verifiable tasks, multi-turn agent tasks rarely provide a unique ground-truth action at each step, since multiple action sequences may solve the same task. Therefore, a self-distillation teacher cannot simply rely on a standard answer to evaluate intermediate actions. We instead construct teacher-only privileged information from transition feedback: each executed action produces a transition (s_t, a_t, s_{t+1}), and we aggregate action-consequence pairs from the same or similar decision contexts. This allows the teacher to judge the student's action in hindsight by comparing the consequences of alternative actions, without requiring a unique correct action.

For an on-policy decision step (s_t^i, a_t^i), we retrieve transitions from $\mathcal{G}$ whose states match the current decision context, denoted by $s \simeq s_t^i$. The resulting teacher-side privileged information is defined as

$$f_t^i = \left\{ (a^{(k)}, s'^{(k)}) \mid s^{(k)} \simeq s_t^i \right\}_{k=1}^{K}, \tag{6}$$

where $a^{(k)}$ is an action executed under a matching decision context and $s'^{(k)}$ is the resulting next state. The relation $s \simeq s_t^i$ can be implemented by exact matching in structured environments or semantic matching in open-ended environments.

The privileged information f_t^i is not a standard answer or expert action label. Instead, it provides hindsight evidence about what different actions have led to in comparable contexts. Thus, the student predicts actions only from the current state, $P_S^\theta(a_t^i \mid s_t^i)$, while the hindsight teacher additionally conditions on transition-level privileged information, $P_T^\theta(a_t^i \mid s_t^i, f_t^i)$. Since f_t^i is constructed after rollouts, it is provided only to the teacher during training, preventing hindsight leakage while keeping the student's training and evaluation inputs consistent.

Self-distilled action quality score. Given the hindsight teacher, we evaluate the quality of the student's sampled action by comparing its token-level probability under the teacher and student distributions. Since each action is generated autoregressively, we define this score at the token level. For rollout i, interaction step t, and token position k, the raw teacher-student discrepancy is

$$d_{t,k}^i = \log P_T^\theta \left(a_{t,k}^i \mid s_t^i, f_t^i, a_{t,<k}^i\right) - \log P_S^\theta \left(a_{t,k}^i \mid s_t^i, a_{t,<k}^i\right). \tag{7}$$

Here, $a_{t,k}^i$ denotes the k-th token of the generated action a_t^i, and $a_{t,<k}^i$ denotes the preceding action tokens. The discrepancy $d_{t,k}^i$ measures how the hindsight teacher reassesses the sampled token after observing transition-level privileged information. A positive value indicates that the token becomes more favorable under the teacher's hindsight evaluation, while a negative value indicates that it becomes less favorable.

To obtain a bounded action quality score, we apply a $\tanh$ transformation $g_{t,k}^i = \tanh(d_{t,k}^i)$. The resulting score $g_{t,k}^i \in (-1, 1)$ provides a bounded estimate of the token's local action quality under hindsight teacher evaluation. This score reflects how the hindsight teacher revises the credit of each generated token. We use it to refine the original reward-derived advantage instead of replacing it, following the residual-correction view in Appendix C.2. Proposition 1 and Corollary 1 further show that such correction can improve token-level credit estimation when it is aligned with the missing local credit and its magnitude is properly controlled.

Entropy-aware confidence calibration. The self-distilled action quality score should not be trusted equally across all tokens. When the teacher distribution is concentrated, the hindsight evaluation is

more confident; when it is flat, the evaluation is more uncertain. We therefore use the entropy of the teacher's token-level distribution to calibrate the strength of $g^i_{t,k}$.

For each token position, let

$$q^i_{t,k}(v) = P^\theta_T\left(v \mid s^i_t, f^i_t, a^i_{t,<k}\right), \quad v \in \mathcal{V}, \tag{8}$$

where $\mathcal{V}$ denotes the vocabulary. The teacher entropy is defined as

$$H^i_{t,k} = -\sum_{v\in\mathcal{V}} q^i_{t,k}(v)\log q^i_{t,k}(v). \tag{9}$$

We convert entropy into a confidence weight by $\tilde{w}^i_{t,k} = \exp(-H^i_{t,k})$, so lower-entropy teacher predictions receive larger weights. To keep the overall scale stable, we normalize the weights over all valid action tokens $w^i_{t,k} = \frac{\tilde{w}^i_{t,k}}{\frac{1}{|\Omega|}\sum_{(j,u,l)\in\Omega}\tilde{w}^j_{u,l}}$, where $\Omega = \left\{(j,u,l) : a^j_{u,l} \text{ is a valid action token}\right\}$.

Finally, we define the entropy-aware self-distilled action quality score as $\bar{g}^i_{t,k} = \lambda w^i_{t,k} g^i_{t,k}$, where λ controls the overall strength of the hindsight teacher. The resulting score $\bar{g}^i_{t,k}$ preserves the teacher's token-level quality judgment while calibrating its influence according to teacher confidence. This calibration controls the relative allocation and magnitude of self-distilled corrections. Corollary 1 shows that a useful correction can still hurt credit estimation when its magnitude is too large. Appendix C.5 further shows that the normalized entropy weights, together with the bounded $\tanh$ gate, bound the correction scale and prevent the self-distilled signal from arbitrarily dominating the reward-derived advantage.

4.3 Asymmetric Advantage Reshaping

Given the rollout group $\mathcal{G}$, we compute the reward-derived trajectory advantage A^i for each rollout using the group-based normalization in Eq. (1). Since the task reward is only available at the trajectory level, A^i is initially assigned to all valid action tokens, i.e., $A^i_{t,k} = A^i$, providing the outcome-grounded update direction for rollout i. However, this uniform assignment cannot distinguish the contributions of different actions or tokens. SDIAL therefore refines $A^i_{t,k}$ with the entropy-aware self-distilled action quality score $\bar{g}^i_{t,k}$, which provides a local hindsight evaluation of the generated token. Following the residual-correction view in Appendix C.2, SDIAL uses $\bar{g}^i_{t,k}$ to reshape the original advantage rather than replace it.

Positive and negative advantages require different operators, as detailed in Appendix C.6. For $A^i_{t,k} < 0$, the trajectory-level outcome only indicates that the overall trajectory performs poorly. In long-horizon multi-turn tasks, such failure may be caused by later mistakes, so earlier tokens in the same trajectory may still be locally useful. A purely multiplicative rule with a positive modulation factor would keep all these tokens negatively updated and thus cannot correct such coarse penalties. We therefore use an additive correction for negative advantages $\hat{A}^i_{t,k} = A^i_{t,k} + \bar{g}^i_{t,k}$. This form allows the hindsight evaluation to strengthen the penalty for unfavorable tokens or offset the penalty for locally reasonable tokens in failed trajectories.

Overall, the reshaped token-level advantage is defined as

$$\hat{A}^i_{t,k} = \begin{cases} A^i_{t,k}\left(1 + \bar{g}^i_{t,k}\right), & A^i_{t,k} \geq 0, \\ A^i_{t,k} + \bar{g}^i_{t,k}, & A^i_{t,k} < 0. \end{cases} \tag{10}$$

This asymmetric design preserves outcome-grounded positive updates while allowing flexible correction of noisy penalties in failed trajectories. Appendix C.6 further analyzes the sign-preserving property of the positive branch and the role of additive correction for addressing locally useful tokens in negative trajectories.

5 Experiments.

In this section, we report the performance of SDIAL across a variety of agentic tasks, focusing on the following research questions (RQs). **RQ1:** How does SDIAL perform compared to closed-source LLMs and RL baselines? **RQ2:** How does each component of SDIAL contribute to overall performance? **RQ3:** How robust is SDIAL to noisy and sparse rewards?

5.1 EXPERIMENT SETUP

Benchmarks. We evaluate SDIAL on three representative multi-turn agent benchmarks: **ALFWorld** (Shridhar et al., 2020) , **BabyAI** (Chevalier-Boisvert et al., 2018), and **WebShop** (Yao et al., 2022a). ALFWorld is a text-based embodied household environment that requires agents to complete long-horizon tasks through navigation and object manipulation. It provides sparse task-level rewards, making credit assignment over intermediate actions challenging. BabyAI is a grid-world instruction-following benchmark with compositional goals and structured observations. Compared with ALFWorld, BabyAI provides a more controlled setting for evaluating whether the agent can learn fine-grained decision policies under long-horizon sparse supervision. (Dwaracherla et al., 2024; Rocamonde et al., 2023) WebShop is a web-based shopping benchmark where the agent must interact with natural-language product pages, search results, and item descriptions to purchase a target product. Unlike ALFWorld and BabyAI, WebShop is more open-ended, with natural-language observations, rarely repeated decision states, and dynamic action consequences.

Baselines. We benchmark SDIAL against several strong baselines spanning different agent paradigms. (1) **Closed-source LLMs.** We report results from GPT-4o (Achiam et al., 2023) and Gemini-2.5-Pro (Team et al., 2023) as high-performing reference models for general reasoning and decision making. (2) **Prompt-based agents.** We include ReAct (Yao et al., 2022b) and Reflexion (Shinn et al., 2023), which induce multi-step behavior through prompting without updating model parameters. (3) **RL-based training methods.** We compare with PPO (Schulman et al., 2017), a standard actor–critic algorithm with an auxiliary value function, as well as critic-free group-based methods including RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), and GiGPO (Feng et al., 2025). In addition, we include RAGEN-2 (Wang et al., 2026b), a stronger group-based optimization baseline designed for multi-turn agent learning.

Training details. We use Qwen2.5-7B-Instruct (Team et al., 2024) as the base model for all trainable methods and train all RL-based methods with LoRA (Hu et al., 2022) under the same backbone, rollout budget, action interface, and maximum interaction horizon. SDIAL optimizes the student policy with GRPO by replacing the sequence-level advantage with our entropy-aware token-level advantage. The hindsight teacher is dynamically constructed from the current student policy using transition-level privileged information from on-policy rollouts, with fixed and EMA teacher variants evaluated in ablations. For group-based RL methods, the rollout group size is set to $N = 8$. Full training details are provided in Appendix B. Detailed training-cost profiling and compute-matched evaluations are provided in Appendix B.2.

5.2 OVERALL PERFORMANCE OF SDIAL (RQ1)

As shown in Table 1 and Table 2, SDIAL achieves the strongest aggregate performance across the three evaluated environments. On ALFWorld, SDIAL attains the highest overall success rate of **90.9%**, outperforming the GiGPO appendix reproduction (86.6%) by **4.3** percentage points and its GRPO backbone (75.5%) by **15.4** points. At the task level, SDIAL achieves the best performance on Look, Clean, Heat, and Pick2, whereas RAGEN-2 performs best on Pick and GiGPO performs best on Cool. On BabyAI, SDIAL likewise obtains the highest overall success rate of **63.5%**, exceeding GiGPO by **1.5** percentage points and Gemini-2.5-Pro (60.1%) by **3.4** percentage points. In particular, SDIAL outperforms GiGPO on *go to*, *open and open*, and *pick up*, while achieving comparable performance on *open* (87.0% versus 87.4%). These results suggest that transition-based self-distilled credit signals provide additional learning signals beyond conventional reward-based step-level credit assignment, leading to consistent gains across diverse task types.

On the more open-ended WebShop benchmark, SDIAL achieves the best overall performance, with a score of **86.2** and a success rate of **79.7%**. Its score surpasses the strongest non-SDIAL result, GRPO (83.8), by **2.4** points, while its success rate exceeds that of GiGPO (72.7%) by **7.0** percentage points. The improvement on WebShop is particularly notable because the environment involves natural-language observations, highly diverse decision states, and action outcomes that depend on dynamically changing page contexts. Taken together, SDIAL achieves the strongest aggregate results on ALFWorld, BabyAI, and WebShop among the evaluated methods. The consistent improvements across these heterogeneous environments provide empirical evidence that transition-based self-distillation can effectively complement both trajectory-level supervision and reward-based step-level credit assignment.

Table 1: Performance comparison on ALFWorld. Values are success rates (%). Results are averaged over 3 random seeds. Mean±std are computed over multiple runs.

Type	Method	ALFWorld						
		Pick	Look	Clean	Heat	Cool	Pick2	All
Closed-Source Model								
Prompting	GPT-4o	75.3	60.8	31.2	56.7	21.6	49.8	48.0
Prompting	Gemini-2.5-Pro	92.8	63.3	62.1	69.0	26.6	58.7	60.3
Qwen2.5-7B-Instruct								
Prompting	Qwen2.5	33.4	21.6	19.3	6.9	2.8	3.2	14.8
Prompting	ReAct	48.5	35.4	34.3	13.2	18.2	17.6	31.2
Prompting	Reflexion	62.0	41.6	44.9	30.9	36.3	23.8	42.7
RL Training	PPO (with critic)	$85.2_{\pm 5.4}$	$84.0_{\pm 3.9}$	$95.3_{\pm 4.2}$	$75.3_{\pm 4.1}$	$61.4_{\pm 4.6}$	$70.8_{\pm 7.2}$	$78.9_{\pm 4.0}$
RL Training	RLOO	$85.9_{\pm 4.0}$	$66.7_{\pm 8.4}$	$91.1_{\pm 6.3}$	66.2 ± 2.4	$65.4_{\pm 11.3}$	$58.0_{\pm 4.0}$	$74.2_{\pm 4.1}$
RL Training	GRPO	$83.8_{\pm 5.5}$	$70.1_{\pm 5.9}$	$89.8_{\pm 5.0}$	$68.5_{\pm 11.5}$	$69.3_{\pm 9.5}$	$61.9_{\pm 6.6}$	$75.5_{\pm 3.0}$
RL Training	RAGEN-2	$\mathbf{96.7}_{\pm 0.1}$	$71.8_{\pm 4.4}$	$92.0_{\pm 6.9}$	$46.7_{\pm 15.3}$	$71.6_{\pm 2.1}$	$57.1_{\pm 4.5}$	$76.8_{\pm 1.6}$
RL Training	GiGPO	$91.7_{\pm 2.8}$	$78.9_{\pm 4.1}$	$93.5_{\pm 1.6}$	$89.4_{\pm 4.0}$	$\mathbf{75.6}_{\pm 4.8}$	$88.8_{\pm 3.7}$	$86.6_{\pm 1.2}$
RL Training	**SDIAL**	$91.0_{\pm 2.0}$	$\mathbf{93.3}_{\pm 11.5}$	$\mathbf{98.9}_{\pm 1.9}$	$\mathbf{98.2}_{\pm 3.0}$	$73.9_{\pm 8.7}$	$\mathbf{90.0}_{\pm 5.0}$	$\mathbf{90.9}_{\pm 2.4}$

Table 2: Performance comparison on BabyAI and WebShop. Values are success rates (%) for BabyAI and score/success rate for WebShop. Results are averaged over 3 random seeds. Mean±std are computed over multiple runs.

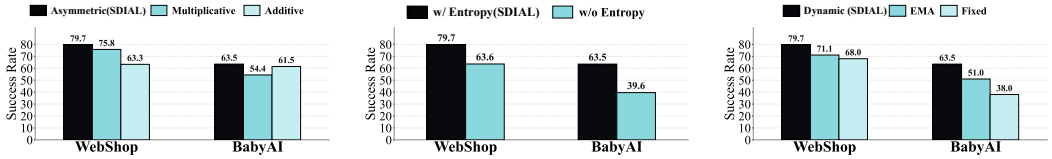
Type	Method	BabyAI					WebShop	
		go to	open and open	pick up	open	All	Score	Succ.
Closed-Source Model								
Prompting	GPT-4o	81.2	0.0	38.9	64.0	48.0	31.8	23.7
Prompting	Gemini-2.5-Pro	86.7	16.8	39.5	84.4	60.1	42.5	35.9
Qwen2.5-7B-Instruct								
Prompting	Qwen2.5	23.2	0.0	5.9	15.0	14.1	26.4	7.8
Prompting	ReAct	30.4	0.0	15.6	25.0	18.8	46.2	19.5
Prompting	Reflexion	33.3	0.0	17.1	30.0	22.7	58.1	28.8
RL Training	PPO (with critic)	$78.9_{\pm 6.1}$	$4.9_{\pm 5.3}$	$17.7_{\pm 6.1}$	$80.9_{\pm 9.4}$	$48.2_{\pm 5.2}$	$80.1_{\pm 2.0}$	$68.8_{\pm 4.0}$
RL Training	RLOO	$71.9_{\pm 13.4}$	$6.1_{\pm 3.8}$	$15.6_{\pm 3.1}$	$73.5_{\pm 8.7}$	$44.0_{\pm 6.1}$	$79.0_{\pm 2.6}$	$65.9_{\pm 4.8}$
RL Training	GRPO	$74.9_{\pm 5.5}$	$1.9_{\pm 3.2}$	$21.8_{\pm 1.5}$	$67.4_{\pm 10.7}$	$46.4_{\pm 3.9}$	$83.8_{\pm 1.9}$	$67.7_{\pm 1.2}$
RL Training	RAGEN-2	$70.2_{\pm 1.2}$	$3.3_{\pm 5.8}$	$24.2_{\pm 10.0}$	$68.2_{\pm 8.0}$	$45.6_{\pm 1.6}$	$78.6_{\pm 1.3}$	$69.0_{\pm 1.2}$
RL Training	GiGPO	$87.5_{\pm 3.9}$	$18.2_{\pm 5.0}$	$31.0_{\pm 9.1}$	$\mathbf{87.4}_{\pm 4.4}$	$62.0_{\pm 2.4}$	$82.1_{\pm 1.7}$	$72.7_{\pm 0.8}$
RL Training	**SDIAL**	$\mathbf{88.0}_{\pm 1.3}$	$\mathbf{23.1}_{\pm 6.1}$	$40.5_{\pm 11.5}$	$87.0_{\pm 11.6}$	$\mathbf{63.5}_{\pm 1.6}$	$\mathbf{86.2}_{\pm 3.2}$	$\mathbf{79.7}_{\pm 3.9}$

5.3 Ablation Studies (RQ2)

As shown in Figure 2, we conduct ablation studies on BabyAI and WebShop to understand the contribution of each component in SDIAL. We compare SDIAL with variants that ablate or modify its key components. We also study the sensitivity to the self-distillation coefficient; detailed results are provided in Appendix B.4.

Effect of the asymmetric advantage reshaping strategy. We compare SDIAL with two variants, *Additive* and *Multiplicative*, which apply the self-distilled signal additively or multiplicatively to all samples without considering the sign of the original advantage. As shown in Figure 2a , SDIAL achieves the strongest final performance on both WebShop and BabyAI. The *Additive* variant can benefit from direct correction but is more prone to late-stage instability, as noisy teacher signals may overly affect reward-supported positive updates. In contrast, *Multiplicative* better preserves the original update direction, but its correction becomes weak when the trajectory-level advantage is small. By applying multiplicative modulation to positive advantages and additive correction to negative advantages, SDIAL combines the stability of sign-preserving updates with the flexibility of correcting noisy trajectory-level penalties, leading to more stable and effective learning.

Effect of entropy-aware signal reweighting. We evaluate entropy-aware reweighting by comparing SDIAL with a *w/o entropy* variant that applies the self-distilled signal with a fixed weight. As shown in Figure 2b, SDIAL achieves better final performance on both WebShop and BabyAI, with



(a) Advantage reshaping (b) w/o Entropy study (c) Teacher update strategy

Figure 2: Ablation studies of SDIAL on BabyAI and WebShop. We evaluate the effects of different advantage reshaping strategies, entropy-aware reweighting and teacher update strategies.

Table 3: Robustness comparison under noisy and sparse rewards. We report final success rates (%) and performance drops from the normal reward setting. Smaller drops indicate stronger robustness.

Method	Normal	Noise $p = 0.1$	Sparsity $q = 0.9$	Sparsity $q = 0.7$
GRPO	67.7	53.1 ($\downarrow$ 21.6%)	57.8 ($\downarrow$ 14.6%)	34.5 ($\downarrow$ 49.0%)
SDIAL	79.7	68.8 ($\downarrow$ 13.7%)	73.7 ($\downarrow$ 7.5%)	54.9 ($\downarrow$ 31.1%)

clearer gains and fewer late-stage fluctuations on WebShop. These results indicate that entropy-based weighting suppresses uncertain teacher guidance and stabilizes advantage reshaping.

Comparison of different teacher update strategies. We compare three teacher update strategies for generating self-distilled credit signals: *Dynamic*, *EMA*, and *Fixed* teachers. The dynamic teacher is updated with the student policy, the EMA teacher tracks a smoothed but delayed version of the student, and the fixed teacher remains unchanged after initialization. As shown in Figure 2c, the dynamic teacher used in SDIAL achieves the best final performance, while EMA and fixed teachers converge to lower success rates. A fixed teacher may become outdated as the student improves, while an EMA teacher provides more stable guidance but may lag behind the current policy. This suggests that teacher signals should remain aligned with the evolving student policy during training.

5.4 Robustness under Noisy and Sparse Rewards (RQ3)

We further evaluate SDIAL under two challenging reward settings: noisy outcome rewards and sparse positive rewards. For reward noise, we randomly flip the final binary reward with probability $p = 0.1$:

$$r' = \begin{cases} 1 - r, & \text{with probability } p, \\ r, & \text{otherwise.} \end{cases}$$

For reward sparsity, we downsample positive rewards during training by keeping each successful trajectory reward with probability $q = 0.9$ and setting the remaining successful rewards to 0. This makes positive feedback less frequent and further increases the difficulty of credit assignment.

As shown in Table 3, SDIAL consistently outperforms GRPO under noisy and sparse rewards. With reward noise $p = 0.1$, SDIAL achieves 68.8%, exceeding GRPO by 15.7 points and showing a smaller relative drop (13.7% vs. 21.6%). Under reward sparsity, SDIAL also remains more robust: it reaches 73.7% at $q = 0.9$ and 54.9% at $q = 0.7$, outperforming GRPO by 15.9 and 20.4 points, respectively. Its performance degradation is also consistently smaller, suggesting that self-distilled credit signals help SDIAL better tolerate unreliable or sparse trajectory-level feedback.

6 Conclusions

We proposed SDIAL, a self-distilled advantage reshaping framework for improving credit assignment in multi-turn agentic reinforcement learning. By using state transitions as hindsight feedback, SDIAL converts teacher-student distributional discrepancies into entropy-aware token-level credit signals and integrates them through asymmetric advantage reshaping, refining coarse trajectory-level rewards while preserving reward-grounded policy optimization. Experiments on ALFWorld, BabyAI, and WebShop show that SDIAL achieves the strongest aggregate result on all three benchmarks among the evaluated methods. Ablation and robustness studies further validate the roles of asymmetric reshaping, entropy-aware weighting, and dynamic teacher updates.

AI USE STATEMENT

Generative AI tools were used for limited assistance with software development, translation, literature search, reference formatting, and language editing. They also supported the interpretation and presentation of experimental results. No reported experimental measurement was generated or modified using these tools. All AI-assisted material was reviewed and, where necessary, revised by the authors, who retain full responsibility for the methodology, empirical claims, and final manuscript.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 225–237, 2024.

Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):9737–9757, 2024.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. *arXiv preprint arXiv:1810.08272*, 2018.

Vikranth Dwaracherla, Seyed Mohammad Asghari, Botao Hao, and Benjamin Van Roy. Efficient exploration for llms. *arXiv preprint arXiv:2402.00396*, 2024.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Xue Jiang, Yihong Dong, Mengyang Liu, Hongyi Deng, Tian Wang, Yongding Tao, Rongyu Cao, Binhua Li, Zhi Jin, Wenpin Jiao, et al. Coderl+: Improving code generation via reinforcement with execution semantics alignment. *arXiv preprint arXiv:2510.18471*, 2025.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. 2024.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Junbo Li, Peng Zhou, Rui Meng, Meet P Vadera, Lihong Li, and Yang Li. Turn-ppo: Turn-level advantage estimation with ppo for improved multi-turn rl in agentic llms. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6227–6243, 2026.

Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning. *arXiv preprint arXiv:2310.12921*, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Minah Seo, Luiz Felipe Vecchietti, Sangkeum Lee, and Dongsoo Har. Rewards prediction-based credit assignment for reinforcement learning with sparse binary rewards. *IEEE Access*, 7:118776–118791, 2019.

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.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

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.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*, 2025.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Qwen Team et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3), 2024.

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*, 2026a.

Peijie Wang, Zhong-Zhi Li, Fei Yin, Dekang Ran, and Cheng-Lin Liu. Mv-math: Evaluating multimodal math reasoning in multi-visual contexts. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 19541–19551, 2025.

Zihan Wang, Chi Gui, Xing Jin, Qineng Wang, Licheng Liu, Kangrui Wang, Shiqi Chen, Linjie Li, Zhengyuan Yang, Pingyue Zhang, et al. Ragen-2: Reasoning collapse in agentic rl. *arXiv preprint arXiv:2604.06268*, 2026b.

Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, et al. Reinforcing multi-turn reasoning in llm agents via turn-level reward design. *arXiv preprint arXiv:2505.11821*, 2025.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Yajie Yang, Wenxiang Chen, Zhihao Zhang, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, et al. Agentprm: Process reward models for llm agents via step-wise promise and progress. *arXiv preprint arXiv:2511.08325*, 2025.

Yibo Yan, Jiamin Su, Jianxiang He, Fangteng Fu, Xu Zheng, Yuanhuiyi Lyu, Kun Wang, Shen Wang, Qingsong Wen, and Xuming Hu. A survey of mathematical reasoning in the era of multimodal large language model: Benchmark, method & challenges. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11798–11827, 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.

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, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022b.

Zihao Yi, Jiarui Ouyang, Zhe Xu, Yuwen Liu, Tianhao Liao, Haohao Luo, and Ying Shen. A survey on recent advances in llm-based multi-turn dialogue systems. *ACM Computing Surveys*, 58(6): 1–38, 2025.

Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, Yang Katie Zhao, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via turn-level credit assignment. In *ICML 2025 Workshop on Computer Use Agents*, 2025.

Jixiao Zhang and Chunsheng Zuo. Grpo-lead: A difficulty-aware reinforcement learning approach for concise mathematical reasoning in language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5642–5665, 2025.

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.

Yaoyu Zhu, Di Huang, Hanqi Lyu, Xiaoyun Zhang, Chongxiao Li, Wenxuan Shi, Yutong Wu, Jianan Mu, Jinghua Wang, Yang Zhao, et al. Qimeng-codev-r1: Reasoning-enhanced verilog generation. *arXiv preprint arXiv:2505.24183*, 2025.

Algorithm 1: SDIAL Policy Optimization

Input: Student policy P_S^θ, environment $\mathcal{E}$, rollout group size G, distillation coefficient λ
Output: Optimized student policy P_S^θ
for *each training iteration* **do**
  Sample a rollout group $\mathcal{G} = \{\tau^i\}_{i=1}^G$ using P_S^θ as in Eq. (5);
  Compute reward-derived trajectory advantages $\{A^i\}_{i=1}^G$ using Eq. (1);
  Assign the trajectory-level advantage to all valid action tokens, $A_{t,k}^i = A^i$;
  for *each rollout i and each interaction step t* **do**
    Construct teacher-only privileged information f_t^i from matched transition consequences using Eq. (6);
    for *each valid action token $a_{t,k}^i$* **do**
      Compute the teacher–student discrepancy $d_{t,k}^i$ using Eq. (7);
      Transform $d_{t,k}^i$ into the self-distilled action quality score $g_{t,k}^i$;
      Compute the teacher entropy $H_{t,k}^i$ using Eqs. (8)–(9) and the entropy-based confidence weight $w_{t,k}^i$;
      Obtain the entropy-aware self-distilled action quality score $\bar{g}_{t,k}^i$ from $g_{t,k}^i$ and $w_{t,k}^i$;
      Reshape the token-level advantage and obtain $\hat{A}_{t,k}^i$ using Eq. (10);
  Update P_S^θ by optimizing the policy objective with reshaped advantages $\hat{A}_{t,k}^i$;

A Additional Method Details

A.1 Overall Implementation

Algorithm 1 summarizes the overall training procedure of SDIAL. At each iteration, the student policy first samples an on-policy rollout group and computes reward-derived trajectory advantages, which are initially assigned to all valid action tokens. For each decision step, we retrieve matched transition consequences from the rollout group to construct the teacher-only privileged information f_t^i. The hindsight teacher is implemented dynamically by evaluating the current policy under this augmented context, while the student is evaluated only under the original decision state. We stop gradients through the teacher branch, so the teacher is used only to compute the self-distilled action quality score, and the policy update is applied through the student likelihood.

In practice, we compute the action quality score only on valid generated action tokens, excluding prompt, observation, and padding tokens. The raw teacher–student discrepancy is bounded with $\tanh(\cdot)$, and the entropy-based confidence weights are normalized within each training batch to keep the scale of the reshaped advantages stable. When fewer than K matched transitions are available, we use all available matches rather than introducing unrelated transitions. For positive advantages, the multiplicative factor is clipped to remain positive, preventing reward-supported updates from being reversed. After obtaining the reshaped token-level advantages, we optimize the student with the same clipped policy objective, replacing the original token advantage with $\hat{A}_{t,k}^i$. At inference time, only the student policy is used; no hindsight teacher or transition retrieval is required.

A.2 Extend to Open-ended Environments

In the main method, SDIAL constructs teacher-side privileged information f_t^i by retrieving action-consequence pairs from the same or similar decision contexts. In structured environments, this retrieval can be implemented by exact state matching. However, in open-ended environments such as WebShop, observations are usually expressed in natural language and the same decision state rarely appears with identical surface forms. Therefore, exact matching may retrieve too few comparable transitions.

To handle this case, we instantiate the matching relation $s \simeq s_t^i$ with similarity-based retrieval. We maintain a transition pool

$$\mathcal{D} = \{(s_m, a_m, s'_m)\}_{m=1}^M,$$

Table 4: WebShop sensitivity to the transition-pool size. The reported similarity is the average lexical similarity of retained transitions.

Method	K	Similarity (%)	Success (%)
GRPO	–	–	67.7 ± 1.2
SDIAL	1	N/A	69.8 ± 3.8
SDIAL	5	89.1	73.2 ± 2.6
SDIAL	8	80.2	79.7 ± 3.9
SDIAL	10	69.6	66.8 ± 2.5

where each tuple records a previously observed state, the executed action, and the resulting next state. For the current decision state s_t^i, we retrieve the top-K transitions whose source states are most similar to s_t^i:

$$\mathcal{C}(s_t^i) = \left\{ (a_m, s'_m) \mid m \in \mathrm{TopK}_m \, \mathrm{Sim}\left(\phi(s_m), \phi(s_t^i)\right) \right\}, \tag{11}$$

where $\phi(\cdot)$ denotes a state encoder and $\mathrm{Sim}(\cdot, \cdot)$ denotes the similarity function. In our WebShop implementation, we compare raw textual states using lexical sequence similarity, restrict candidates to the same task identifier, and discard candidates whose similarity is below 0.5. The current action-consequence pair is retained as anchor evidence whenever it is available.

The retrieved action-consequence pairs are used only to construct the teacher-side privileged information f_t^i. The student policy still conditions only on the original state s_t^i, and no retrieval module is used during inference. When fewer than K matched transitions are available, we use all available transitions. If the current transition is unavailable and no retrieved candidate passes the threshold, SDIAL falls back to the original reward-derived advantage for that step. This fallback occurs in 0.1% of decision steps in the reported WebShop analysis. Thus, the open-ended extension only modifies the retrieval procedure for constructing f_t^i, while the self-distilled action quality scoring and asymmetric advantage reshaping remain unchanged.

B Experiment Details

B.1 Details of Training

Hyperparameters for ALFWorld. For all experiments, we adopt Qwen2.5-7B-Instruct as the backbone model and keep the core training hyperparameters identical across all reinforcement learning algorithms to ensure a fair comparison. Specifically, the maximum prompt length is set to 2048 tokens, and the maximum response length is set to 512 tokens. The actor learning rate is fixed to 5×10^{-5}. For all group-based RL methods, we use a group size of 8 and sample 16 different groups per rollout, resulting in a total of 16 × 8 = 128 environments. Each episode allow a maximum of 50 training steps. All model weights are loaded in *safetensors* format. LoRA is applied uniformly across all RL methods. The LoRA rank and scaling factor are set to 64 and 64, respectively, and the same LoRA configuration is used unless otherwise specified. **SDIAL-Specific Settings:** For SDIAL, we set `sd_teacher_mode=dynamic`, where the teacher is updated together with the student policy to provide on-policy self-distilled credit signals. The distillation coefficient is set to `sd_distill_coeff=0.2`, controlling the strength of the self-distilled signal in advantage reshaping. We also enable entropy-aware signal reweighting with `sd_use_entropy_weighting=true`, so that teacher predictions with lower entropy receive stronger weights. These settings are used by default, and alternative choices are evaluated in the ablation studies.

Hyperparameters for WebShop. For all experiments, we adopt Qwen2.5-7B-Instruct as the backbone model and keep the core training hyperparameters identical across all reinforcement learning algorithms to ensure a fair comparison. Specifically, the maximum prompt length is set to 4096 tokens, and the maximum response length is set to 512 tokens. The actor learning rate is fixed to 5×10^{-5}. For WebShop, the group size is 8 and the rollout uses 96 environments in total. Each episode allows a maximum of 15 training steps. All model weights are loaded in *safetensors* format. LoRA is applied uniformly across all RL methods. The LoRA rank and scaling factor are set to 64 and 64, respectively, and the same LoRA configuration is used unless otherwise

specified. **SDIAL-Specific Settings:** For SDIAL, we set `sd_teacher_mode=dynamic`, where the teacher is updated together with the student policy to provide on-policy self-distilled credit signals. The distillation coefficient is set to `sd_distill_coeff=0.2`, controlling the strength of the self-distilled signal in advantage reshaping. We also enable entropy-aware signal reweighting with `sd_use_entropy_weighting=true`, so that teacher predictions with lower entropy receive stronger weights. These settings are used by default, and alternative choices are evaluated in the ablation studies.

Hyperparameters for BabyAI. For all experiments, we adopt Qwen2.5-7B-Instruct as the backbone model and keep the core training hyperparameters identical across all reinforcement learning algorithms to ensure a fair comparison. Specifically, the maximum prompt length is set to 2048 tokens, and the maximum response length is set to 512 tokens. The actor learning rate is fixed to 5×10^{-5}. For all group-based RL methods, we use a group size of 8 and sample 16 different groups per rollout, resulting in a total of 16 × 8 = 128 environments. Each episode allows a maximum of 30 training steps. All model weights are loaded in *safetensors* format. LoRA is applied uniformly across all RL methods. The LoRA rank and scaling factor are set to 64 and 64, respectively, and the same LoRA configuration is used unless otherwise specified. **SDIAL-Specific Settings:** For SDIAL, we set `sd_teacher_mode=dynamic`, where the teacher is updated together with the student policy to provide on-policy self-distilled credit signals. The distillation coefficient is set to `sd_distill_coeff=1.0`, controlling the strength of the self-distilled signal in advantage reshaping. We also enable entropy-aware signal reweighting with `sd_use_entropy_weighting=true`, so that teacher predictions with lower entropy receive stronger weights. These settings are used by default, and alternative choices are evaluated in the ablation studies.

Computing Details. All experiments on ALFWorld and Webshop and BabyAI are conducted using Qwen2.5-7B-Instruct with LoRA fine-tuning, and each experiment is trained for a total of 150 training steps. The default experimental setup requires only four NVIDIA A800 GPUs, each equipped with 80 GB of GPU memory.

B.2 Training Cost and Compute-matched Evaluation

We report training-cost profiling and compute-matched evaluations separately for each backbone and hardware setting. The hindsight teacher and transition retrieval are used only during training; at inference time, only the student policy is retained.

Qwen2.5-7B-Instruct profiling. We profile PPO, GRPO, and SDIAL on four NVIDIA A800 GPUs under the same training configuration. Relative to GRPO, SDIAL increases time per epoch by 12.51%, peak memory by 4.86%, and average memory by 8.29%. The additional cost is primarily due to the training-only hindsight teacher forward pass.

Table 5: Training-cost profiling on Qwen2.5-7B-Instruct using four NVIDIA A800 GPUs.

Method	Time / epoch (s)	Peak memory (GiB)	Average memory (GiB)
PPO	1437.75	77.73	60.06
GRPO	501.26	70.81	40.39
SDIAL	563.96	74.25	43.74

Qwen3-4B WebShop profiling and wall-clock matching. For WebShop, the Qwen3-4B profiling uses the same hardware and training configuration for GRPO and SDIAL. SDIAL increases time per step by 19.86%, peak memory by 0.59%, and average memory by 2.56%.

Table 6: Qwen3-4B WebShop training-cost profiling.

Method	Time / step (s)	Peak memory (GiB)	Average memory (GiB)
GRPO	511.38	78.85	47.72
SDIAL	612.93	79.31	48.95
Relative change	+19.86%	+0.59%	+2.56%

Table 7: Qwen3-4B WebShop compute-matched evaluation. Results are averaged over three random seeds.

Budget	Method	Steps	Success (%)
Step-matched	GRPO	100	34.12 ± 3.25
Wall-clock-matched (17h)	GRPO	120	45.45 ± 3.46
Wall-clock-matched (17h)	SDIAL	100	53.65 ± 5.20

Under the same approximately 17-hour wall-clock budget, SDIAL outperforms GRPO by 8.20 percentage points even though GRPO completes more updates.

Qwen3-4B ALFWorld profiling and wall-clock matching. For ALFWorld, the comparison uses Qwen3-4B on eight NVIDIA H800 GPUs. SDIAL increases time per epoch by 16.97%, peak memory by 13.36%, and average memory by 4.33% relative to GRPO.

Table 8: Qwen3-4B ALFWorld training-cost profiling.

Method	Time / epoch (s)	Peak memory (GiB)	Average memory (GiB)
GRPO	711.64	69.78	43.45
SDIAL	832.38	79.11	45.33
Relative change	+16.97%	+13.36%	+4.33%

Table 9: Qwen3-4B ALFWorld wall-clock-matched evaluation on 8 NVIDIA H800 GPUs. Each seed is evaluated on 128 tasks.

Method	Updates	Time	Pick	Look	Clean	Heat	Cool	Pick2	Overall
GRPO	130	24.13 h	75.31 ± 5.66	58.33 ± 8.33	56.08 ± 3.82	44.13 ± 2.20	32.22 ± 5.14	41.26 ± 3.81	52.08 ± 1.80
SDIAL	111	24.31 h	82.72 ± 4.28	58.33 ± 0.00	66.98 ± 10.16	83.65 ± 4.42	53.25 ± 9.64	43.73 ± 3.90	**64.32** ± 3.16

GRPO completes 130 updates while SDIAL completes 111 updates, but the wall-clock difference is only 10.8 minutes (0.75%). SDIAL remains 12.24 percentage points higher overall, providing a compute-matched comparison on a second environment.

B.3 Prompts

Figure 3, Figure 4, and Figure 5 show the task prompts used for ALFWorld, BabyAI, and WebShop, respectively. Figure 6, Figure 7, and Figure 8 show the corresponding teacher prompts used to construct hindsight evaluation signals.

We use prompt templates for three environments: ALFWorld, WebShop, and BabyAI. For each environment, we design two prompt variants: a student prompt and a teacher prompt. The student prompt contains the information normally available to the agent at decision time, while the teacher prompt augments the student prompt with additional transition evidence obtained from parallel exploration.

Prompt Template for ALFWorld

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

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

Figure 3: Prompt used for ALFWorld.

Prompt Template for BabyAI

You are an expert agent operating in a BabyAI gridworld environment.
Your mission is: {tasks}
Now you are in Step: {current_step} ,you have taken {step_count} steps

Recent actions (last {history_length} steps):
{action_history}

current observation is {current_observation}

Available actions: left, right, forward, pickup, drop, toggle

Now it's your turn to take an action.
You should first choose one available action for the current step and present it within <action> </action> tags.
After the action, explain the reasoning that supports it within <think> </think> tags.

Figure 4: Prompt used for BabyAI.

All prompt templates are constructed using Python-style string formatting. Placeholders enclosed in curly braces, such as `{task_description}`, `{current_step}`, `{step_count}`, `{action_history}`, `{current_observation}`, and `{available_actions}`, are dynamically populated at runtime. These placeholders represent the task instruction, the current step index, the number of executed steps, recent action-observation history, the current observation, and the admissible actions, respectively.

For ALFWorld and WebShop, the student prompt provides the task description, recent interaction history, the current observation, and the admissible action space. The agent is instructed to first reason about the current situation and then select one admissible action. The reasoning process is enclosed within `<think>` and `</think>` tags, and the final action decision is enclosed within `<action>` and `</action>` tags. For BabyAI, the student prompt follows the same decision-making format, but the admissible action space is fixed as `left`, `right`, `forward`, `pickup`, `drop`, and `toggle`.

The teacher prompt contains all information in the corresponding student prompt, but additionally includes a *Parallel Exploration Evidence* block. This block provides action-conditioned one-step transition evidence collected from parallel exploration at the same state. Each evidence item consists of a candidate action and the immediate next observation caused by executing that action:

```
Evidence 1: Action: {action1}
→ Immediate next observation: {observation1}
```

Here, `{action1}` denotes one candidate action executed from the current state, and `{observation1}` denotes the immediate observation returned by the environment after taking that

Prompt Template for WebShop

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

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

Figure 5: Prompt used for WebShop.

Teacher Prompt Template for ALFWorld

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

=== Parallel Exploration Evidence ===
At this exact state, the following action-conditioned transitions were observed across 1 parallel trajectories.
Use these immediate observations as evidence about what each action does from this state.
Do not copy an action only because it appears in the evidence list; choose based on the current state and the transition evidence.
Evidence 1: Action： {action1}
→ Immediate next observation:{observation1}
Evidence 2: Action： {action2}
→ Immediate next observation:{observation2}
Evidence 3: Action： {action3}
→ Immediate next observation:{observation3}
....

=== End Parallel Exploration Evidence ===
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 6: Teacher Prompt used for ALFWorld.

action. In other words, `Action 1` and `Observation 1` form a one-step transition pair: if the agent takes `Action 1` at the current state, the environment immediately produces `Observation 1` as the next observation. The same interpretation applies to `Action 2` and `Observation 2`, `Action 3` and `Observation 3`, and so on.

B.4 LAMBDA SENSITIVITY STUDY

We study the effect of the self-distillation coefficient λ, which controls the strength of the entropy-aware self-distilled signal. As shown in Figure 9, the preferred value is environment-dependent:

Teacher Prompt Template for BabyAI

You are an expert agent operating in a BabyAI gridworld environment.
Your mission is: {tasks}
Now you are in Step: {current_step} ,you have taken {step_count} steps

Recent actions (last {history_length} steps):
{action_history}

current observation is {current_observation}

Available actions: left, right, forward, pickup, drop, toggle

=== Parallel Exploration Evidence ===
At this exact state, the following action-conditioned transitions were observed across 1 parallel trajectories.
Use these immediate observations as evidence about what each action does from this state.
Do not copy an action only because it appears in the evidence list; choose based on the current state and the transition evidence.
Evidence 1: Action : {action1}
→ Immediate next observation:{observation1}
Evidence 2: Action : {action2}
→ Immediate next observation:{observation2}
Evidence 3: Action : {action3}
→ Immediate next observation:{observation3}
....

=== End Parallel Exploration Evidence ===
Now it's your turn to take an action.
You should first choose one available action for the current step and present it within <action> </action> tags.
After the action, explain the reasoning that supports it within <think> </think> tags.

Figure 7: Teacher Prompt used for BabyAI.

$\lambda = 0.2$ performs better on WebShop, where teacher signals may be noisier in open-ended interactions, while $\lambda = 1.0$ performs better on the more structured BabyAI benchmark. These results suggest that the coefficient balances reward-derived advantages and self-distilled credit signals, and should be tuned according to the reliability of teacher guidance.

C Conditional Analysis of SDIAL Credit Reshaping

This appendix provides a conditional analysis of the credit reshaping mechanism in SDIAL. We do not claim that SDIAL unconditionally dominates GRPO in all possible settings. Instead, we show that SDIAL yields a more accurate token-level credit estimator when the self-distilled correction is sufficiently aligned with the residual token-level credit omitted by the sequence-level GRPO advantage, and when the correction magnitude is controlled.

This analysis should be interpreted as a sufficient-condition result rather than a universal guarantee. The key question is whether the hindsight teacher provides a correction signal that is informative about local credit variations within a trajectory. Under this condition, SDIAL can be viewed as a residual credit correction to GRPO.

C.1 Credit Assignment View of GRPO

Consider a valid action token indexed by $(i, t, k) \in \Omega$, where i denotes the rollout index, t denotes the interaction step, and k denotes the token position within the generated action. As defined in Eq. (1), GRPO computes a reward-derived trajectory-level advantage A^i. This trajectory-level advantage is uniformly assigned to all valid action tokens in the same trajectory:

$$A^i_{t,k} = A^i, \qquad (i, t, k) \in \Omega. \tag{12}$$

Teacher Prompt Template for WebShop

You are an expert autonomous agent operating in the WebShop e-commerce environment.
Your task is to: {task_description}.
Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took: {action_history}
You are now at step {current_step} and your current observation is: {current_observation}.
Your admissible actions of the current situation are:
[
{available_actions}
].
=== Parallel Exploration Evidence ===
At this exact state, the following action-conditioned transitions were observed across 1 parallel trajectories.
Use these immediate observations as evidence about what each action does from this state.
Do not copy an action only because it appears in the evidence list; choose based on the current state and the transition evidence.
Evidence 1: Action： {action1}
→ Immediate next observation:{observation1}
Evidence 2: Action： {action2}
→ Immediate next observation:{observation2}
Evidence 3: Action： {action3}
→ Immediate next observation:{observation3}
....

=== End Parallel Exploration Evidence ===
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 8: Teacher Prompt used for WebShop.

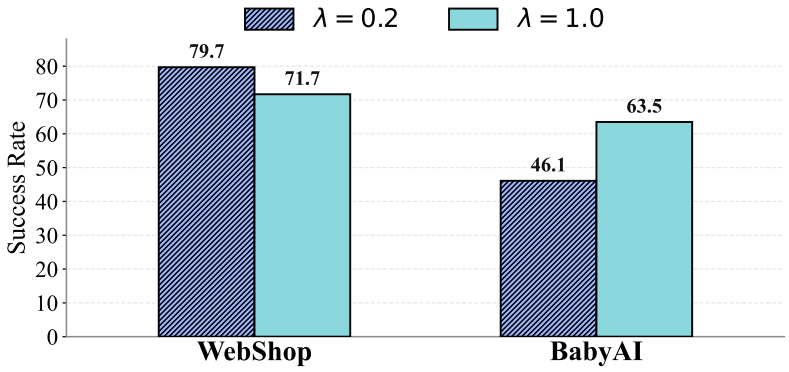


Figure 9: Self-distillation coefficient sensitivity on BabyAI and WebShop.

Therefore, GRPO can be viewed as using the following token-level credit estimator:

$$\hat{C}^{\mathrm{GRPO}}_{i,t,k} = A^i_{t,k}. \tag{13}$$

To make the oracle token-level credit precise, we view autoregressive action generation as a token-level decision process. Let

$$h^i_{t,k} = \left(s^i_t, a^i_{t,<k}\right) \tag{14}$$

denote the token-level decision context, which consists of the environment state and the previously generated action tokens. We define the oracle token-level credit as the policy advantage of selecting token $a^i_{t,k}$ under this token-level context:

$$C^i_{t,k} = A^\pi\left(h^i_{t,k}, a^i_{t,k}\right) = Q^\pi\left(h^i_{t,k}, a^i_{t,k}\right) - V^\pi\left(h^i_{t,k}\right). \tag{15}$$

In sparse-reward multi-turn tasks, this oracle credit is generally unavailable, because the environment only provides delayed trajectory-level feedback. GRPO therefore uses the same sequence-level advantage A^i as a coarse approximation for every token in trajectory i.

However, different tokens in the same trajectory may contribute differently to the final outcome. Some tokens may be locally useful even if the overall trajectory fails, while some tokens may be locally harmful even if the trajectory succeeds. To formalize the token-level credit variation omitted by GRPO, we define the residual token-level credit as

$$r^i_{t,k} = C^i_{t,k} - A^i_{t,k}. \tag{16}$$

When $r^i_{t,k} > 0$, GRPO underestimates the local contribution of token $a^i_{t,k}$. When $r^i_{t,k} < 0$, GRPO overestimates its local contribution. Thus, improving over GRPO can be understood as estimating the residual credit $r^i_{t,k}$ ignored by the uniform sequence-level advantage.

C.2 SDIAL as Residual Credit Correction

SDIAL keeps the reward-derived GRPO advantage and introduces a token-wise self-distilled correction. According to the asymmetric advantage reshaping rule in Eq. (10), SDIAL produces the reshaped advantage

$$\hat{A}^i_{t,k} = \begin{cases} A^i_{t,k}\left(1 + \bar{g}^i_{t,k}\right), & A^i_{t,k} \geq 0, \\ A^i_{t,k} + \bar{g}^i_{t,k}, & A^i_{t,k} < 0. \end{cases} \tag{17}$$

Equivalently, we can write SDIAL as

$$\hat{C}^{\mathrm{SDIAL}}_{i,t,k} = \hat{A}^i_{t,k} = A^i_{t,k} + \Delta^i_{t,k}, \tag{18}$$

where $\Delta^i_{t,k}$ is the SDIAL correction term:

$$\Delta^i_{t,k} = \begin{cases} A^i_{t,k}\bar{g}^i_{t,k}, & A^i_{t,k} \geq 0, \\ \bar{g}^i_{t,k}, & A^i_{t,k} < 0. \end{cases} \tag{19}$$

This formulation shows that SDIAL does not replace the reward-derived GRPO advantage. Instead, it refines $A^i_{t,k}$ by adding a token-wise residual correction. If $\Delta^i_{t,k}$ is close to the residual credit $r^i_{t,k} = C^i_{t,k} - A^i_{t,k}$, then the reshaped advantage

$$A^i_{t,k} + \Delta^i_{t,k}$$

will be closer to the oracle token-level credit $C^i_{t,k}$ than the original GRPO estimate $A^i_{t,k}$.

C.3 Sufficient Condition for Improved Token-level Credit Estimation

We measure the quality of a token-level credit estimator by its expected squared error to the oracle token-level credit. This choice is intentional. The sequence-level reward advantage used by GRPO is directly grounded in observed trajectory returns, but it provides coarse supervision when uniformly assigned to all tokens in long-horizon trajectories. In contrast, SDIAL introduces a self-distilled correction, which may introduce bias if the teacher signal is inaccurate, but can reduce token-level credit error when the correction is informative. Therefore, the expected squared error provides a unified way to capture the overall deviation from an oracle token-level credit:

$$\mathcal{E}(\hat{C}) = \mathbb{E}\left[\sum_{(i,t,k)\in\Omega}\left(\hat{C}_{i,t,k} - C^i_{t,k}\right)^2\right]. \tag{20}$$

This metric is used only for theoretical comparison. It does not require the oracle token-level credit to be available during training.

Proposition 1 (Sufficient condition for improved credit assignment)**.** *If the SDIAL correction* $\Delta^i_{t,k}$ *satisfies*

$$2\mathbb{E}\left[\sum_{(i,t,k)\in\Omega}\Delta^i_{t,k}r^i_{t,k}\right] > \mathbb{E}\left[\sum_{(i,t,k)\in\Omega}\left(\Delta^i_{t,k}\right)^2\right], \tag{21}$$

then SDIAL has lower token-level credit estimation error than GRPO:

$$\mathcal{E}(\hat{C}^{\mathrm{SDIAL}}) < \mathcal{E}(\hat{C}^{\mathrm{GRPO}}). \tag{22}$$

Proof. For GRPO, using Eq. (13), the estimation error of token (i,t,k) is

$$\hat{C}^{\mathrm{GRPO}}_{i,t,k} - C^i_{t,k} = A^i_{t,k} - C^i_{t,k} = -r^i_{t,k}. \tag{23}$$

Therefore,

$$\mathcal{E}(\hat{C}^{\mathrm{GRPO}}) = \mathbb{E}\left[\sum_{(i,t,k)\in\Omega}\left(r^i_{t,k}\right)^2\right]. \tag{24}$$

For SDIAL, using Eq. (18), the estimation error becomes

$$\hat{C}^{\mathrm{SDIAL}}_{i,t,k} - C^i_{t,k} = A^i_{t,k} + \Delta^i_{t,k} - C^i_{t,k} = \Delta^i_{t,k} - r^i_{t,k}. \tag{25}$$

Thus,

$$\mathcal{E}(\hat{C}^{\mathrm{SDIAL}}) = \mathbb{E}\left[\sum_{(i,t,k)\in\Omega}\left(\Delta^i_{t,k} - r^i_{t,k}\right)^2\right]. \tag{26}$$

Taking the difference between Eq. (26) and Eq. (24), we obtain

$$\begin{aligned}
&\mathcal{E}(\hat{C}^{\mathrm{SDIAL}}) - \mathcal{E}(\hat{C}^{\mathrm{GRPO}}) \\
&= \mathbb{E}\left[\sum_{(i,t,k)\in\Omega}\left(\left(\Delta^i_{t,k} - r^i_{t,k}\right)^2 - \left(r^i_{t,k}\right)^2\right)\right] \\
&= \mathbb{E}\left[\sum_{(i,t,k)\in\Omega}\left(\left(\Delta^i_{t,k}\right)^2 - 2\Delta^i_{t,k}r^i_{t,k}\right)\right] \\
&= \mathbb{E}\left[\sum_{(i,t,k)\in\Omega}\left(\Delta^i_{t,k}\right)^2\right] - 2\mathbb{E}\left[\sum_{(i,t,k)\in\Omega}\Delta^i_{t,k}r^i_{t,k}\right].
\end{aligned} \tag{27}$$

Therefore, if Eq. (21) holds, then

$$\mathcal{E}(\hat{C}^{\mathrm{SDIAL}}) - \mathcal{E}(\hat{C}^{\mathrm{GRPO}}) < 0,$$

which proves

$$\mathcal{E}(\hat{C}^{\mathrm{SDIAL}}) < \mathcal{E}(\hat{C}^{\mathrm{GRPO}}).$$

□

Proposition 1 has a direct interpretation. The term

$$\mathbb{E}\left[\sum_{(i,t,k)\in\Omega}\Delta^i_{t,k}r^i_{t,k}\right]$$

measures how well the SDIAL correction aligns with the true residual credit omitted by GRPO. The term

$$\mathbb{E}\left[\sum_{(i,t,k)\in\Omega}\left(\Delta^i_{t,k}\right)^2\right]$$

measures the magnitude of the correction. Thus, SDIAL improves over GRPO when the correction is sufficiently aligned with the missing token-level credit and is not too large. This matches the design goal of SDIAL: the self-distilled signal should refine the sequence-level advantage rather than overwhelm it.

C.4 Residual Alignment of the Hindsight Teacher

We next connect the sufficient condition in Proposition 1 to the hindsight self-distillation mechanism in SDIAL. Recall that SDIAL constructs a hindsight teacher using transition-level privileged information f_t^i, which aggregates action-consequence pairs from the same or similar decision contexts. This information is not an expert action label. Instead, it provides hindsight evidence about what different actions have led to in comparable states.

The entropy-aware self-distilled signal is

$$\bar{g}_{t,k}^i = \lambda w_{t,k}^i g_{t,k}^i.$$

To separate the correction direction from its overall scale, define the unscaled correction direction

$$z_{t,k}^i = \begin{cases} A_{t,k}^i w_{t,k}^i g_{t,k}^i, & A_{t,k}^i \geq 0, \\ w_{t,k}^i g_{t,k}^i, & A_{t,k}^i < 0. \end{cases} \tag{28}$$

Then the SDIAL correction in Eq. (19) can be written as

$$\Delta_{t,k}^i = \lambda z_{t,k}^i. \tag{29}$$

To make the effect of asymmetric reshaping explicit, we decompose the valid token set into positive- and negative-advantage subsets:

$$\Omega_+ = \{(i,t,k) \in \Omega : A_{t,k}^i \geq 0\}, \qquad \Omega_- = \{(i,t,k) \in \Omega : A_{t,k}^i < 0\}. \tag{30}$$

We define the alignment contributions on the two subsets as

$$\mu_+ = \mathbb{E}\left[\sum_{(i,t,k)\in\Omega_+} z_{t,k}^i r_{t,k}^i\right], \qquad \mu_- = \mathbb{E}\left[\sum_{(i,t,k)\in\Omega_-} z_{t,k}^i r_{t,k}^i\right], \tag{31}$$

and their second moments as

$$\nu_+ = \mathbb{E}\left[\sum_{(i,t,k)\in\Omega_+} \left(z_{t,k}^i\right)^2\right], \qquad \nu_- = \mathbb{E}\left[\sum_{(i,t,k)\in\Omega_-} \left(z_{t,k}^i\right)^2\right]. \tag{32}$$

Let

$$\mu = \mu_+ + \mu_-, \qquad \nu = \nu_+ + \nu_-. \tag{33}$$

Assumption 1 (Decomposed residual alignment)**.** *The self-distilled correction direction has positive overall alignment with the residual token-level credit:*

$$\mu = \mu_+ + \mu_- > 0, \tag{34}$$

with finite second moment

$$\nu = \nu_+ + \nu_- < \infty. \tag{35}$$

A stronger but more interpretable sufficient condition is $\mu_+ \geq 0$, $\mu_- \geq 0$, and at least one of them is strictly positive.

Assumption 1 does not require the hindsight teacher to be perfect. It only requires the teacher-induced correction to be positively correlated, in expectation, with the residual token-level credit ignored by GRPO. The decomposition into Ω_+ and Ω_- makes explicit that the two asymmetric branches may contribute differently. For positive-advantage tokens, the teacher signal should mainly redistribute the strength of reward-supported updates. For negative-advantage tokens, the teacher signal should help distinguish locally harmful tokens from locally reasonable tokens inside failed trajectories.

The assumption is motivated by the construction of the hindsight teacher. If a sampled token is locally beneficial but under-credited by the trajectory-level advantage, the transition-conditioned teacher is expected to assign higher probability to this token than the student, producing a positive discrepancy $d_{t,k}^i$ and hence a positive bounded signal $g_{t,k}^i$. If a sampled token is locally harmful but over-credited by GRPO, the teacher is expected to assign lower probability to this token, producing a negative discrepancy and hence a negative bounded signal. Thus, the self-distilled signal can be viewed as a noisy estimator of the residual credit direction.

Importantly, this residual-alignment assumption is not guaranteed by construction. Rather, it captures the condition under which hindsight self-distillation provides useful local credit information. In practice, this condition can be empirically supported by ablation studies and correlation analyses that examine whether the teacher-induced correction is associated with local hindsight outcomes or downstream performance improvements.

Corollary 1 (Correction strength condition)**.** *Under Assumption 1, if the correction strength satisfies*

$$0 < \lambda < \frac{2\mu}{\nu}, \tag{36}$$

then SDIAL has lower token-level credit estimation error than GRPO.

Proof. Substituting $\Delta^i_{t,k} = \lambda z^i_{t,k}$ into the sufficient condition in Eq. (21), we need

$$2\lambda \mathbb{E}\left[\sum_{(i,t,k)\in\Omega} z^i_{t,k} r^i_{t,k}\right] > \lambda^2 \mathbb{E}\left[\sum_{(i,t,k)\in\Omega} \left(z^i_{t,k}\right)^2\right]. \tag{37}$$

Using the decomposition in Eqs. (31)–(33), this becomes

$$2\lambda(\mu_+ + \mu_-) > \lambda^2(\nu_+ + \nu_-), \tag{38}$$

or equivalently,

$$2\lambda\mu > \lambda^2\nu. \tag{39}$$

Since $\lambda > 0$, this inequality holds whenever

$$0 < \lambda < \frac{2\mu}{\nu}.$$

Then Proposition 1 implies

$$\mathcal{E}(\hat{C}^{\mathrm{SDIAL}}) < \mathcal{E}(\hat{C}^{\mathrm{GRPO}}).$$

□

Corollary 1 explains why the correction strength λ is important. Even if the self-distilled signal is directionally useful, an overly large correction may overshoot the residual credit and increase the estimation error. Thus, SDIAL requires the correction to be both informative and controlled.

C.5 Effect of Entropy-aware Reweighting

The entropy-aware weight $w^i_{t,k}$ modulates the self-distilled signal according to teacher confidence. In the main text, SDIAL first computes the unnormalized confidence score

$$\tilde{w}^i_{t,k} = \exp(-H^i_{t,k}),$$

and then normalizes it over all valid token positions:

$$w^i_{t,k} = \frac{\tilde{w}^i_{t,k}}{\frac{1}{|\Omega|}\sum_{(j,u,l)\in\Omega} \tilde{w}^j_{u,l}}. \tag{40}$$

This normalization keeps the average confidence weight equal to one:

$$\frac{1}{|\Omega|}\sum_{(i,t,k)\in\Omega} w^i_{t,k} = 1. \tag{41}$$

Therefore, entropy reweighting changes the relative strength of token-level corrections without changing their average scale.

For any two valid token positions (i,t,k) and (j,u,l), the relative confidence weight is

$$\frac{w^i_{t,k}}{w^j_{u,l}} = \frac{\exp(-H^i_{t,k})}{\exp(-H^j_{u,l})} = \exp\left(-\left(H^i_{t,k} - H^j_{u,l}\right)\right). \tag{42}$$

Hence, if

$$H^i_{t,k} < H^j_{u,l},$$

then

$$w^i_{t,k} > w^j_{u,l}.$$

Tokens for which the hindsight teacher is more confident receive larger relative corrections, while tokens with higher teacher entropy are relatively down-weighted.

The bounded gate also limits the raw teacher-student discrepancy:

$$\left|g^i_{t,k}\right| = \left|\tanh(d^i_{t,k})\right| \le 1. \tag{43}$$

Since the normalized weights are positive and averaged to one, they are finite within each rollout group. Let

$$W_{\max} = \max_{(i,t,k)\in\Omega} w^i_{t,k}.$$

Then

$$\left|\bar{g}^i_{t,k}\right| = \left|\lambda w^i_{t,k} g^i_{t,k}\right| \le \lambda W_{\max}. \tag{44}$$

In practice, the normalized confidence weights or the final modulation factor can also be clipped for numerical stability, which directly controls $W_{\max}$.

Consequently, the SDIAL correction satisfies

$$\left|\Delta^i_{t,k}\right| \le \begin{cases} \lambda W_{\max}\left|A^i_{t,k}\right|, & A^i_{t,k} \ge 0, \\ \lambda W_{\max}, & A^i_{t,k} < 0. \end{cases} \tag{45}$$

This boundedness prevents the self-distilled signal from arbitrarily dominating the reward-derived GRPO advantage.

Importantly, entropy-aware reweighting should be interpreted as a confidence-based mechanism for allocating correction strength, rather than a guarantee of teacher correctness. A low-entropy teacher distribution indicates a confident teacher prediction, but it does not by itself ensure that the teacher signal is correct. Thus, the analysis here establishes relative weighting and scale control; the usefulness of entropy-aware reweighting still depends on whether lower teacher entropy correlates with more reliable correction signals in practice.

C.6 Role of Asymmetric Reshaping

We finally analyze the role of the asymmetric reshaping rule.

For positive-advantage tokens, SDIAL uses multiplicative modulation:

$$\hat{A}^i_{t,k} = A^i_{t,k}\left(1 + \bar{g}^i_{t,k}\right), \qquad A^i_{t,k} \ge 0. \tag{46}$$

A positive advantage indicates that the sampled trajectory should be encouraged according to the reward signal. Therefore, SDIAL uses the self-distilled signal conservatively as a modulation factor. The sign-preserving property holds when the modulation factor is clipped to remain positive, or under the sufficient condition $\lambda W_{\max} < 1$:

$$1 + \bar{g}^i_{t,k} > 0. \tag{47}$$

Thus, for $A^i_{t,k} > 0$,

$$\operatorname{sign}\left(\hat{A}^i_{t,k}\right) = \operatorname{sign}\left(A^i_{t,k}\right). \tag{48}$$

This means that SDIAL preserves the reward-supported positive update direction while allowing the teacher signal to redistribute update strength across tokens.

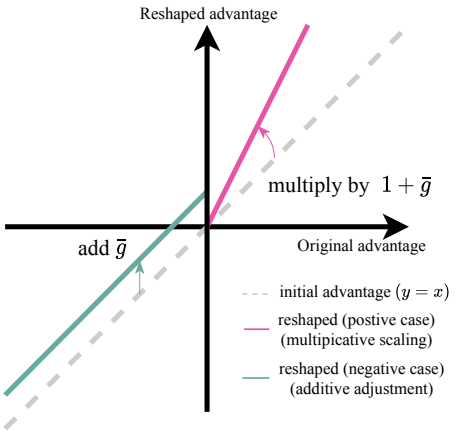


Figure 10: Sign-conditioned advantage refinement for positive and negative advantages.

For negative-advantage tokens, SDIAL uses additive correction:

$$\hat{A}^i_{t,k} = A^i_{t,k} + \bar{g}^i_{t,k}, \qquad A^i_{t,k} < 0. \tag{49}$$

This design is intended to address a common credit-assignment issue in long-horizon multi-turn tasks. A trajectory may receive negative reward because of later mistakes, but not every earlier token is necessarily harmful. Under GRPO, all tokens in this trajectory are uniformly suppressed. In contrast, SDIAL can distinguish tokens within the failed trajectory.

If

$$\bar{g}^i_{t,k} < 0,$$

then the token receives stronger suppression. If

$$\bar{g}^i_{t,k} > 0,$$

then the penalty is weakened. Moreover, when

$$\bar{g}^i_{t,k} > \left|A^i_{t,k}\right|, \tag{50}$$

we have

$$A^i_{t,k} + \bar{g}^i_{t,k} > 0. \tag{51}$$

Thus, a locally useful token can receive positive credit even when the overall trajectory has negative advantage.

This sign correction is impossible for a purely multiplicative rule with a positive modulation factor, because multiplying a negative advantage by a positive factor cannot change its sign. Therefore, the additive rule for negative advantages can mitigate the "good-token-in-bad-trajectory" problem when the correction is sufficiently aligned with local credit. Under the residual-alignment assumption, this mechanism allows SDIAL to correct noisy sequence-level penalties using hindsight information. These observations are conditional and do not establish global optimality, unbiasedness, or universal superiority of the sign-conditioned operator.

C.7 Connection to Policy-gradient Estimation

The analysis above focuses on token-level credit estimation. We now connect it to the policy-gradient estimator used in training.

Let

$$\psi^i_{t,k} = \nabla_\theta \log P^\theta_S\left(a^i_{t,k} \mid s^i_t, a^i_{t,<k}\right) \tag{52}$$

be the token-level score function. The oracle token-level policy-gradient estimator is

$$G^* = \sum_{(i,t,k)\in\Omega} C^i_{t,k}\psi^i_{t,k}. \tag{53}$$

A practical estimator using a credit estimator $\hat{C}_{i,t,k}$ is

$$\hat{G} = \sum_{(i,t,k)\in\Omega} \hat{C}_{i,t,k}\psi^i_{t,k}. \tag{54}$$

The gradient estimation error is therefore

$$\hat{G} - G^* = \sum_{(i,t,k)\in\Omega} \left(\hat{C}_{i,t,k} - C^i_{t,k}\right)\psi^i_{t,k}. \tag{55}$$

Assume that the token-level score functions are bounded:

$$\left\|\psi^i_{t,k}\right\| \le L, \tag{56}$$

and that the number of valid action tokens is bounded by M, i.e.,

$$|\Omega| \le M. \tag{57}$$

Then by Cauchy's inequality,

$$\begin{aligned}\left\|\hat{G}-G^*\right\|^2 &= \left\|\sum_{(i,t,k)\in\Omega}\left(\hat{C}_{i,t,k}-C^i_{t,k}\right)\psi^i_{t,k}\right\|^2 \\ &\le \left(\sum_{(i,t,k)\in\Omega}\left(\hat{C}_{i,t,k}-C^i_{t,k}\right)^2\right)\left(\sum_{(i,t,k)\in\Omega}\left\|\psi^i_{t,k}\right\|^2\right) \\ &\le L^2|\Omega|\sum_{(i,t,k)\in\Omega}\left(\hat{C}_{i,t,k}-C^i_{t,k}\right)^2 \\ &\le L^2M\sum_{(i,t,k)\in\Omega}\left(\hat{C}_{i,t,k}-C^i_{t,k}\right)^2. \end{aligned} \tag{58}$$

Taking expectation on both sides gives

$$\mathbb{E}\left[\left\|\hat{G}-G^*\right\|^2\right] \le L^2M\mathcal{E}(\hat{C}). \tag{59}$$

Therefore, a lower token-level credit estimation error leads to a tighter upper bound on the policy-gradient estimation error under the bounded-score and bounded-token assumptions. Combined with Proposition 1, this shows that SDIAL can provide a more accurate policy-gradient estimator when its self-distilled correction is aligned with the residual token-level credit.

This result should be interpreted as an error-bound perspective on policy-gradient estimation rather than a direct variance-reduction guarantee. It does not imply a strict ordering of the actual gradient errors, since the Cauchy-based bound may be loose. A direct comparison of gradient variance between SDIAL and GRPO would require stronger assumptions on the covariance structure of token-level score functions and on the randomness introduced by the teacher signal. Therefore, our analysis focuses on token-level credit estimation error and its induced upper bound on policy-gradient estimation error.

C.8 Summary

This appendix formalizes SDIAL as a residual credit correction to GRPO. GRPO assigns the same trajectory-level advantage A^i to every token in trajectory i, thereby ignoring token-level credit variation. SDIAL preserves this reward-derived sequence-level signal while adding an entropy-aware self-distilled correction $\Delta^i_{t,k}$. If this correction is positively aligned with the residual token-level credit

$$r^i_{t,k} = C^i_{t,k} - A^i_{t,k}$$

and its magnitude is controlled, SDIAL achieves lower token-level credit estimation error than GRPO. Entropy-aware reweighting modulates the correction strength according to teacher confidence, and asymmetric reshaping preserves reward-supported positive updates while allowing correction of falsely penalized tokens in negative trajectories. This provides a sufficient-condition analysis of when the finer-grained credit assignment mechanism of SDIAL can improve token-level credit estimation.