OVERCOMING SKILL CREDIT BLINDNESS: SKILL-CONDITIONED POLICY OPTIMIZATION FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning reusable reasoning skills through reinforcement learning (RL) is essential for improving both training efficiency and the ability of large language models (LLMs) to solve challenging reasoning tasks. However, most existing methods rely on Group Relative Policy Optimization (GRPO), whose group-relative baseline can absorb systematic skill effects, obscuring whether performance gains arise from policy improvement or injected skills. We term this credit-assignment failure **skill credit blindness**. To address it, we propose **Skill-Conditioned Policy Optimization (SCPO)**, which explicitly conditions policy optimization on skills and reconstructs the advantage signal for reliable attribution. To isolate policy quality from skill effects, SCPO normalizes rewards only within the same problem and skill condition. It then uses paired skill-conditioned and skill-free rollouts to estimate each skill's instance-level marginal contribution, which modulates the advantage to reinforce beneficial skills and suppress harmful ones. Finally, SCPO computes cross-problem skill-level gains to support the evolution of the skill library. Experiments with Qwen3-series models across multiple mathematical reasoning benchmarks show that SCPO consistently outperforms GRPO, demonstrating skill-based RL depends not only on learning reusable skills, but also on correctly attributing skill contributions during policy optimization.

1 INTRODUCTION

With the growing incorporation of tools (Qian et al., 2025) and reusable reasoning skills (Xia et al., 2026) into large language model (LLM) post-training, skill learning has become a promising research problem (Yao et al., 2022; Schick et al., 2023; Paranjape et al., 2023). Beyond invoking external skills, an LLM must determine which skills are genuinely useful, when they should be applied and which skills should be revised or discarded. Existing skill learning methods typically guide exploration with skill labels, experience snippets or reasoning templates, but estimate skill utility from observed success rates, reward differences or usage frequency. Such observational signals are easily confounded by the skill-selection process, since a policy may invoke skills on easier problems, avoid them on harder ones, or repeatedly reuse historically preferred skills. Reliable skill learning therefore requires estimating counterfactual utility, namely how the expected reward would change if only the injected skill were altered.

Group Relative Policy Optimization (GRPO) provides a lightweight way to perform repeated exploration over multiple candidate trajectories to reinforce higher-reward rollouts without training an additional critic model (Shao et al., 2024; Xia et al., 2026). In skill-based RL, an LLM retrieves or generates a skill, injects it into the prompt, samples several solutions and updates the policy according to the relative rewards of these rollouts (Vishe et al., 2026). However, this design implicitly treats the injected skill as part of context rather than as an intervention whose contribution must be separately identified. As a result, the same scalar reward is used to credit both the quality of the reasoning trajectory and the usefulness of the skill condition, which leads to a fundamental credit-assignment problem. GRPO assumes that trajectories within the same group are conditionally comparable, so that reward differences mainly reflect reasoning quality rather than systematic effects from external conditions. But this assumption is often violated in skill-based RL. For the same problem, some trajectories may be generated with a helpful skill, some with no skill and others with a misleading

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.

or irrelevant skill. Using a single group baseline to normalize these heterogeneous trajectories conflates behavioral quality with the effects of skill augmentation. Consequently, the benefit of a useful skill may be canceled by mixed-group normalization, while a harmful skill may be reinforced when it co-occurs with easier problems or strong underlying reasoning trajectories. More importantly, observational reward differences cannot determine whether a skill is genuinely useful, because skill usage is selected by the policy and confounded by problem difficulty, retrieval bias and historical preference. Therefore, the policy update cannot reliably distinguish whether higher reward comes from better reasoning under a fixed skill condition or from the marginal contribution of the skill itself. We refer to this failure mode as **skill credit blindness**.

To address this problem, we propose **Skill-Conditioned Policy Optimization** (SCPO), which decomposes advantage into **behavior credit** and **skill credit**. First, for behavior credit, it replaces problem-level normalization with skill-conditioned group normalization. Rollouts are compared only within the same problem and under the same skill condition, so the advantage reflects relative reasoning quality under a fixed condition rather than reward shifts. Second, for skill credit, SCPO constructs same-problem skill-conditioned and no-skill rollout comparisons to estimate the instance-level skill utility under the current policy. This counterfactual utility is then used by a causal modulation to reshape the advantage update, which can strengthen high-quality trajectories of beneficial skills and attenuate updates for harmful skills. Beyond a single policy update, SCPO accumulates skill-level utility across problems to guide the evolution of the skill library. In this way, the policy learns from skills only when they provide evidence of causal utility, while the library evolves according to how its skills affect downstream rollouts. We evaluate SCPO with Qwen3-series models on multiple mathematical reasoning benchmarks, and the results show consistent improvements over GRPO, indicating that skill-based RL requires not only reusable skills, but also a reliable mechanism for attributing their contributions during optimization.

Our contributions are summarized as follows:

(1) We identify *skill blindness* as a credit-assignment failure of GRPO in skill-based RL and propose *Skill-Conditioned Group Normalization* to restore conditionally valid comparisons among rollouts.

(2) We develop *Causal Advantage Modulation*, which turns same-problem skill/no-skill contrasts into instance-level update weights, enabling policy optimization to reinforce skills by their marginal contribution rather than by their observed reward alone.

(3) We show that SCPO yields consistent gains across mathematical reasoning benchmarks and model scales, with ablations and skill-library analyses confirming that its improvements arise from structured credit decomposition and counterfactual utility-driven skill evolution.

2 Related work

2.1 Reinforcement Learning

Reinforcement learning (RL) has become a central paradigm for large language model (LLM) post-training (Bai et al., 2022). Early work is largely driven by reinforcement learning from human feedback (RLHF), where a learned reward model provides scalar supervision for policy optimization. Proximal Policy Optimization (PPO) (Schulman et al., 2017) is widely used in this setting, but its separate value network introduces substantial memory and compute overhead, and value estimation can be unstable for long-horizon reasoning. As post-training shifts from general alignment toward reasoning-intensive tasks, reinforcement learning with verifiable rewards (RLVR) has become increasingly important, especially for mathematics and code, where rule-based checkers can provide automatic outcome rewards. However, these rewards are often sparse and terminal, making critic-based advantage estimation costly and noisy. To reduce this cost, DeepSeek-Math introduces Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which samples multiple trajectories per problem and derives relative advantages from group-level reward statistics, removing the need for an independent critic. Owing to its simplicity and efficiency, GRPO has been widely adopted in recent reasoning models (Ahmadian et al., 2024; Yu et al., 2025), including DeepSeek-Math (Shao et al., 2024), DeepSeek-R1 (Guo et al., 2025), and Qwen3 (Yang et al., 2025). A common assumption behind these methods is that rollouts within a group are comparable, so that group-normalized advantages reflect policy quality. However, this assumption is not tailored to skill-based RL, where

rollouts for the same problem may differ in skill condition and thus in their underlying reward distribution.

2.2 Skill Learning

As LLMs are applied to more complex reasoning and interactive tasks, recent work has begun to introduce reusable skills to guide exploration and problem solving. Existing skill-based RL methods can be broadly divided into two categories. The first category treats skills as additional conditioning signals for policy optimization, using skill labels, demonstrations or retrieved experiences to reshape the exploration space (Vishe et al., 2026; Wu et al., 2026; Fan et al., 2026). The second category jointly optimizes skill generation and task solving, allowing models to extract reusable strategies from successful trajectories and apply or revise them in later tasks (Li et al., 2026b; Shi et al., 2026; Shah et al., 2026; Xia et al., 2026). While these methods recognize skills as dynamic external assets, they often estimate skill utility from observational signals such as success rate and usage frequency. Such estimates are confounded by natural skill selection: models may invoke skills on easier problems, avoid them on harder ones, or overuse historically favored skills. As a result, observed reward differences can mix task difficulty, policy capability and the effect of the skill.

2.3 Self-Evolving Skill Libraries

In addition to skill-conditioned policy optimization, another line of work studies self-evolving skill libraries. Recent work transforms skill libraries from static repositories into dynamic capability systems, roughly corresponding to different lifecycle stages (Shen et al., 2026; Pan et al., 2026; Yang et al., 2026; Yu et al., 2026; Li et al., 2026a; Zhang et al., 2026; Wei et al., 2026). These studies show that the effectiveness of a skill library depends not only on the quality of individual skills, but also on whether the library can adapt to changing model capabilities and task distributions. However, most lifecycle mechanisms still rely on outcome-based metrics or heuristic rules. A skill may appear useful simply because it is assigned to easier tasks, or appear ineffective because it is invoked only on harder tasks. Thus, maintaining a skill library also requires estimating the causal contribution of each skill, rather than relying solely on observed performance.

3 Problem Formulation

Existing skill-based RL introduces two distinct sources of credit. The first is **behavior credit**: under a fixed problem and skill condition, which trajectory should be reinforced? The second is **skill credit**: for the same problem, does injecting a skill improve the expected outcome relative to not injecting it? A reliable objective should not estimate these two quantities from the same mixed comparison.

For skill credit, the desired quantity is the same-problem gain

$$\Delta(x, s) = \mathbb{E}[R \mid x, \mathrm{do}(S = s)] - \mathbb{E}[R \mid x, \mathrm{do}(S = \varnothing)]. \tag{1}$$

For behavior credit, GRPO constructs a group-relative advantage over rollouts of the same problem:

$$A_i^{\mathrm{GRPO}} = \frac{r_i - \mu_x}{\sigma_x + \epsilon}, \tag{2}$$

where μ_x and σ_x are computed across the rollout group for x. This comparison is valid only when rollouts in the group are conditionally comparable. Skill conditioning breaks this assumption when trajectories generated with different skills, or without skills, are normalized together.

To show the mismatch, decompose the reward as

$$r_i = b(x) + \tau(x, s_i) + \varepsilon_i, \tag{3}$$

where $b(x)$ captures the problem-level reward scale, $\tau(x, s_i)$ is the effect of the skill condition, and ε_i is the trajectory-specific behavior term. Centering by the mixed group mean gives

$$r_i - \mu_x = \underbrace{\tau(x, s_i) - \mathbb{E}_j[\tau(x, s_j)]}_{\text{skill-condition offset}} + \underbrace{\varepsilon_i - \mathbb{E}_j[\varepsilon_j]}_{\text{behavior difference}}. \tag{4}$$

Thus, under mixed skill conditions, the GRPO advantage is no longer a clean estimate of reasoning quality. It is confounded by the skill-condition offset and also reflects which skill condition the rollout was generated under. We refer to this attribution failure as **skill credit blindness**.

4 SKILL-CONDITIONED POLICY OPTIMIZATION

4.1 OVERVIEW

To address the **skill credit blindness** of GRPO, we propose SCPO, which decomposes credit assignment in reinforcement learning into two related but semantically different components: **behavior credit** and **skill credit**. First, it performs skill-conditioned group normalization, comparing rollouts only within the same problem and skill condition, so that the advantage measures behavior credit rather than reward shifts caused by skill usage. Second, it constructs same-problem skill/no-skill controls to estimate the instance-level utility of the injected skill, providing a counterfactual signal for skill credit. Third, it uses causal advantage modulation to combine the two: high-quality trajectories are reinforced more strongly when their skill condition is estimated to be beneficial, and attenuated when the skill is harmful or uncertain. Beyond a single policy update, SCPO aggregates skill utility across problems and training steps to update the external skill library, allowing skills to be refined, promoted, or quarantined as the model changes. Thus, SCPO forms a closed loop between policy optimization and skill-library evolution: the policy learns under skill-conditioned credit assignment, while the skill library evolves according to the counterfactual utility of its skills.

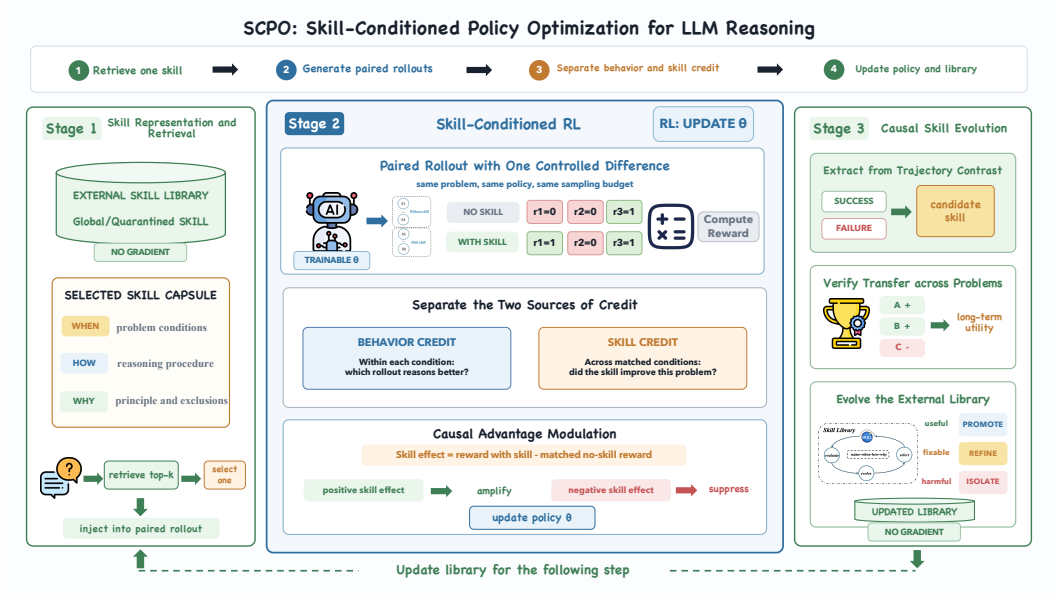

Figure 1: SCPO framework for skill-conditioned credit assignment and skill-library evolution.

4.2 SKILL-CONDITIONED GROUP NORMALIZATION

The role of group normalization in GRPO is to remove problem-level reward scale and compare alternative trajectories for the same problem. This comparison is beneficial only when all trajectories in the group are generated under comparable conditions. However, in skill-based RL, two rollouts for the same problem may differ not only in sampled reasoning paths, but also in the skill condition. If such rollouts share one group baseline, it will absorb skill-induced reward shifts, and the resulting advantage no longer represents behavior quality alone. SCPO restores conditional comparability by defining the comparison group over the composite condition (x, s) rather than over x alone. For a problem x and skill condition s, let

$$\mathcal{G}_{x,s} = \{i \mid x_i = x,\ s_i = s\}. \tag{5}$$

SCPO computes group statistics only inside $\mathcal{G}_{x,s}$:

$$\mu_{x,s} = \frac{1}{|\mathcal{G}_{x,s}|} \sum_{i \in \mathcal{G}_{x,s}} r_i, \qquad \sigma_{x,s} = \mathrm{Std}\left(\{r_i : i \in \mathcal{G}_{x,s}\}\right). \tag{6}$$

The behavior advantage is then

$$A_i^{\text{scgn}} = \frac{r_i - \mu_{x_i,s_i}}{\sigma_{x_i,s_i} + \epsilon}. \tag{7}$$

This estimator removes cross-skill comparisons from the advantage calculation and estimates behavior credit only within a fixed skill condition. Rather than crediting the skill, it asks which trajectory reasons better among rollouts generated for the same problem under the same skill, while the skill's own contribution is estimated separately through paired skill/no-skill comparisons.

4.3 Paired Skill Gain

Skill-conditioned normalization isolates behavior credit, but it cannot decide whether the conditioning skill is useful. A skill may define a coherent comparison group and still be unnecessary or harmful for the problem. Therefore, for each paired evaluation of skill s on problem x, SCPO computes

$$pg_{\text{inst}}(x, s) = \bar{r}(x, s) - \bar{r}(x, \varnothing), \tag{8}$$

where $\bar{r}(x, s)$ and $\bar{r}(x, \varnothing)$ denote the mean rewards of skill-conditioned and no-skill rollouts for the same problem. This instance-level paired gain estimates whether the injected skill improves the current problem under the current policy state. Positive values indicate that the skill contributes beyond the base policy, while negative values indicate that the skill condition is misleading or constraining, and near-zero values indicate that the evidence is insufficient to assign skill credit.

SCPO uses this signal at two timescales. The instance-level gain $pg_{\text{inst}}(x, s)$ modulates the current policy update, preserving sensitivity to the specific problem, while a smoothed skill-level statistic tracks stable utility across problems:

$$\text{PG}_t(s) = (1 - \rho)\text{PG}_{t-1}(s) + \rho\mathbb{E}_{x\in\mathcal{B}_t(s)}\left[pg_{\text{inst}}(x, s)\right], \tag{9}$$

where $\mathcal{B}_t(s)$ is the set of paired evaluations for skill s at stage t. This separation prevents two common errors by keeping long-term averages and individual instances distinct: a skill's long-horizon average should not modulate a mismatched instance, and a single noisy instance should not determine the long-term state of the skill library.

The reward used for paired comparison is based on verifiable correctness, with bounded shaping for stability:

$$r(x, y, s) = r_{\text{acc}}(x, y) + r_{\text{fmt}}(y) + r_{\text{bonus}}(x, y, s), \tag{10}$$

followed by clipping,

$$r(x, y, s) \leftarrow \text{clip}\left(r(x, y, s), r_{\min}, r_{\max}\right). \tag{11}$$

The shaping terms improve the observability of skill effects, while clipping prevents auxiliary rewards from dominating the counterfactual comparison.

4.4 Causal Advantage Modulation

The two methods above target different objects: A_i^{scgn} measures behavior quality under a fixed skill condition, while $pg_{\text{inst}}(x, s)$ measures whether the skill condition is beneficial for the current problem. Then, SCPO combines them through causal advantage modulation, which determines how much behavior credit should enter the policy update given the estimated skill utility.

SCPO maps paired gain to a directional modulation factor:

$$\delta(x, s) = 2\sigma\left(\frac{pg_{\text{inst}}(x, s)}{T}\right), \tag{12}$$

where T is a temperature parameter. This mapping is centered at one, so that zero paired gain leaves the update unchanged, positive paired gain amplifies it, and negative paired gain suppresses it. Therefore, modulation does not reward skill usage. It instead rewards behavior under a skill only when the paired comparison shows that the skill improves the same problem.

To avoid spending update budget on uninformative cases, SCPO further applies an information gate

$$g(x, s) = 4p_{x,s}(1 - p_{x,s}), \tag{13}$$

where $p_{x,s}$ is the empirical success rate under condition (x, s). The gate is largest near the decision boundary and small when the condition is nearly always solved or nearly always failed. The final advantage is

$$\widetilde{A}_i = g(x_i, s_i) \cdot \delta(x_i, s_i) \cdot A_i^{\mathrm{scgn}}. \tag{14}$$

This form makes the update selective in two senses. A rollout must be strong relative to trajectories under the same skill condition, and the skill condition must exhibit positive marginal utility relative to the no-skill control. Thus, SCPO reinforces skill-conditioned reasoning only when both behavior credit and skill credit support the update.

4.5 Optimization Objective

SCPO optimizes the policy with a clipped policy-gradient objective using the modulated advantage. Let $\rho_{i,t}(\theta)$ be the token-level probability ratio and $m_{i,t}$ the response-token mask. The loss is

$$\mathcal{L}_{\mathrm{clip}} = -\mathbb{E}_i \left[\frac{1}{\sum_t m_{i,t}} \sum_t m_{i,t} \min \left(\rho_{i,t}(\theta) \widetilde{A}_i, \mathrm{clip}\left(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right) \widetilde{A}_i \right) \right]. \tag{15}$$

The same trajectory-level advantage is assigned to response tokens in the rollout, following the outcome-reward setting. If an explicit skill-selection policy $q_\phi(s \mid x)$ is trained, SCPO can add a utility-weighted selection objective:

$$\mathcal{L}_{\mathrm{skill}} = -\mathbb{E}_{(x,s)} \left[w(x, s) \log q_\phi(s \mid x) \right], \tag{16}$$

where $w(x, s)$ is derived from the skill-utility signal. This term biases retrieval toward skills with demonstrated counterfactual benefit rather than toward skills that merely co-occur with high rewards.

SCPO also uses an optional diversity-aware reward term for correct trajectories:

$$r_{\mathrm{div}}(x, y_i, s) = \eta_{\mathrm{div}} \cdot \mathbf{1}\left[r_{\mathrm{acc}}(x, y_i) = 1 \right] \cdot \frac{|r(x, y_i, s) - \mu_{x,s}|}{\sigma_{x,s} + \epsilon}. \tag{17}$$

This term is added before advantage estimation. It encourages informative variation within a skill-conditioned group and reduces the dominance of saturated, low-variance groups.

5 Experiments

5.1 Base Model

We use the Qwen3 series as the base policy to evaluate SCPO in a non-saturated reasoning regime. The model is strong enough to produce diverse rollouts required by group-relative optimization, but still leaves substantial room for improvement on challenging mathematical benchmarks. This setting allows us to test whether skill-conditioned credit assignment provides gains beyond GRPO.

5.2 Training Data and Evaluation Benchmarks

The training data consists of Competition Math, the AIME series, and the AMC series (Mathematical Association of America, 2025; Hendrycks et al., 2021). We evaluate models on five mathematical reasoning benchmarks: AIME 2024, AIME 2025, AIME 2026, HMMT 2025, and Omni-math-500. We use Pass@8 as the experimental metric, and all results are evaluated over 10 seeds, with the mean and standard deviation reported.

5.3 Main Results

As shown in Table 1, SCPO improves the average Pass@8 by 2.94 points on Qwen3-4B and 2.70 points on Qwen3-8B, showing that the benefit is not tied to a single model scale. More importantly, the improvement appears in Pass@8, a metric that is sensitive to whether the policy preserves multiple viable solution modes rather than only sharpening its most likely response. This is precisely the regime targeted by SCPO: different skills may expose different reasoning paths, but these paths should be reinforced only when the skill-conditioned rollout is both high quality under its

own condition and supported by a positive same-problem skill effect. By separating behavior credit from skill credit, SCPO changes how rollout evidence shapes the learned sampling distribution. It strengthens skill-conditioned modes that are locally reliable and counterfactually useful, while avoiding indiscriminate reinforcement of every retrieved skill. The main results therefore suggest that SCPO improves skill-based RL by preserving productive reasoning alternatives under repeated sampling, rather than by merely increasing skill usage or adding more exploration.

Table 1: Pass@8 results on mathematical reasoning benchmarks, reported as mean ± standard deviation, with bold indicating the best mean, breaking ties by lower standard deviation.

	Qwen3-4B		Qwen3-8B	
Benchmark	**GRPO**	**SCPO**	**GRPO**	**SCPO**
AIME 2024	$\mathbf{60.0\pm3.3}$	58.0 ± 5.8	61.7 ± 1.7	$\mathbf{65.3\pm2.5}$
AIME 2025	47.3 ± 2.5	$\mathbf{50.7\pm2.5}$	46.0 ± 3.3	$\mathbf{48.3\pm1.7}$
AIME 2026	47.0 ± 4.3	$\mathbf{51.3\pm1.6}$	51.7 ± 1.7	$\mathbf{56.7\pm1.7}$
HMMT 2025	21.0 ± 4.5	$\mathbf{28.7\pm1.6}$	26.7 ± 3.3	$\mathbf{26.7\pm1.6}$
Omni-math-500	42.2 ± 0.5	$\mathbf{43.5\pm0.7}$	43.6 ± 0.5	$\mathbf{46.2\pm0.7}$
Average	43.50	$\mathbf{46.44}$	45.94	$\mathbf{48.64}$

5.4 Ablation Study

As shown in Table 2, removing skill-conditioned normalization weakens the separation between trajectory quality and skill condition. Rollouts generated under different skill states are again compared through a shared group baseline; the resulting bias depends on how different the skill and no-skill reward distributions are on each benchmark, showing that conditioning the comparison set is necessary for making behavior credit well-defined. Removing paired skill gain causes a sharper loss in aggregate performance, which suggests that the key benefit of SCPO is not simply exposing the model to more skill-conditioned trajectories, but identifying which of these trajectories are supported by a genuine same-problem skill effect. Without the paired control, a skill-conditioned rollout can be reinforced because it happens to receive high reward, even if the skill itself did not cause the improvement. Paired gain prevents this observational shortcut by asking whether the same problem improves when the skill is injected relative to a no-skill control. Removing causal modulation further shows that estimating skill utility is insufficient unless it changes the update direction. Skill-conditioned normalization ranks trajectories within a fixed condition, and paired gain estimates whether the condition is useful; modulation is the mechanism that couples these two signals in the policy gradient. It strengthens updates when good trajectories are produced under beneficial skills and attenuates updates when the skill effect is negative or unreliable. The three components are not independent: normalization defines the valid comparison set, paired gain identifies skill credit, and modulation determines how that credit reshapes policy learning.

Table 2: Pass@8 ablation results using Qwen3-8B. Results are reported as mean ± standard deviation. SC, M, and PG denote skill-conditioned group normalization, causal advantage modulation, and paired skill gain, respectively. Bold values indicate the best mean in each row, breaking ties by lower standard deviation.

Benchmark	**Full**	**w/o SC**	**w/o M**	**w/o PG**
AIME 2024	$\mathbf{65.3\pm2.5}$	65.0 ± 4.1	65.3 ± 6.5	58.7 ± 3.0
AIME 2025	48.3 ± 1.7	48.3 ± 4.6	$\mathbf{48.7\pm1.8}$	46.7 ± 4.1
AIME 2026	$\mathbf{56.7\pm1.7}$	56.1 ± 3.3	52.7 ± 2.8	56.7 ± 3.3
HMMT 2025	$\mathbf{26.7\pm1.6}$	25.0 ± 1.8	22.0 ± 1.8	26.0 ± 2.8
Omni-math-500	$\mathbf{46.2\pm0.7}$	43.0 ± 0.7	43.8 ± 1.0	43.6 ± 1.1
Average	$\mathbf{48.64}$	47.48	46.50	46.34

5.5 Analysis of training process

Figure 2 compares the training process of SCPO and GRPO. Both methods improve rapidly at the beginning and then enter a slower optimization regime, but their early trajectories differ. GRPO obtains a higher batch pass rate in the initial steps, while SCPO reaches a higher mean reward after a short warm-up period. This phenomenon is consistent with the additional structure introduced by SCPO. Before skill-conditioned updates become reliable, the policy must explore which skills can be retrieved and executed, and the paired estimator must collect enough evidence to distinguish helpful skills from irrelevant or harmful ones. During this phase, SCPO may sacrifice some early pass-rate efficiency because it does not immediately reinforce all high-reward skill-conditioned rollouts. As training proceeds, however, the pass-rate gap narrows and SCPO matches or modestly exceeds GRPO in later steps, while its mean reward advantage remains more persistent. This suggests that SCPO changes the optimization trajectory rather than simply accelerating reward maximization: it first evaluates heterogeneous skill conditions, then concentrates updates on trajectories supported by positive skill utility. The training curves therefore provide process-level evidence that skill-conditioned credit assignment can improve the quality of policy updates without causing a sustained drop in answer correctness.

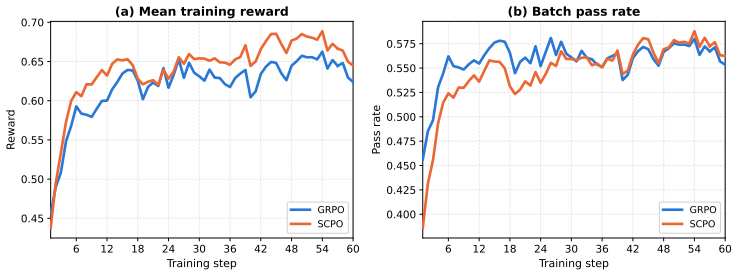

Figure 2: Comparison of the training processes between SCPO and GRPO.

5.6 Skill Library Evolution Analysis

Figure 3 shows that the SCPO skill library evolves from rapid expansion to selective consolidation. In Figure 3(a), the total number of stable skills keeps increasing, while the number of active skills used for injection saturates earlier. This indicates that SCPO does not expose all retained skills to the policy, but filters which skills remain useful for training-time injection. Figures 3(b) and 3(c) show that both skill extraction and improvement acceptance decrease over training, together with the same-domain transfer gain, which indicates early training discovers broad, transferable reasoning patterns, whereas later candidates are more likely to be redundant, narrow, or already internalized by the model. Figure 3(d) further shows quality differentiation inside the library. The share of positive-gain skills decreases, while quarantined skills increase and eventually dominate. This does not imply library degradation; rather, it reflects stricter filtering as the model evolves. Skills that no longer provide marginal benefit, or become harmful under the current policy, are removed from active injection. Overall, the library is not a passive memory buffer, it first expands, then tightens its acceptance criteria, and finally separates active skills from redundant or harmful ones according to long-horizon utility.

5.7 Effect of Advantage Modulation

Figure 4 shows that the modulation layer acts as an instance-dependent credit controller rather than a uniform reward amplifier. Roughly half of the samples receive amplification, about one-third are suppressed, and the rest stay close to neutral (Figure 4(a)), indicating that SCPO does not reinforce all skill-conditioned rollouts indiscriminately. Instead, it selectively preserves positive skill-conditioned updates, attenuates harmful ones, and leaves uncertain cases with little additional influence. This behavior is driven by the instance-level counterfactual signal, where the mean modulation factor δ remains above one but varies over training, closely tracking the temporal changes of pg_{inst} (Figure 4(b)). Thus, modulation is not a fixed reward bonus; it adapts to whether the injected skill improves the current problem relative to its no-skill control. Meanwhile, strongly positive

skill effects consistently outnumber negative ones, although negative cases remain non-negligible throughout training (Figure 4(c)), showing that skills are useful on average but not uniformly reliable across instances. SCPO therefore converts heterogeneous skill utility into asymmetric policy updates. Useful skills receive stronger credit, harmful skills are suppressed, and ambiguous effects contribute only weakly. This provides a mechanism-level explanation for why SCPO can benefit from external skills without reinforcing every retrieved skill indiscriminately.

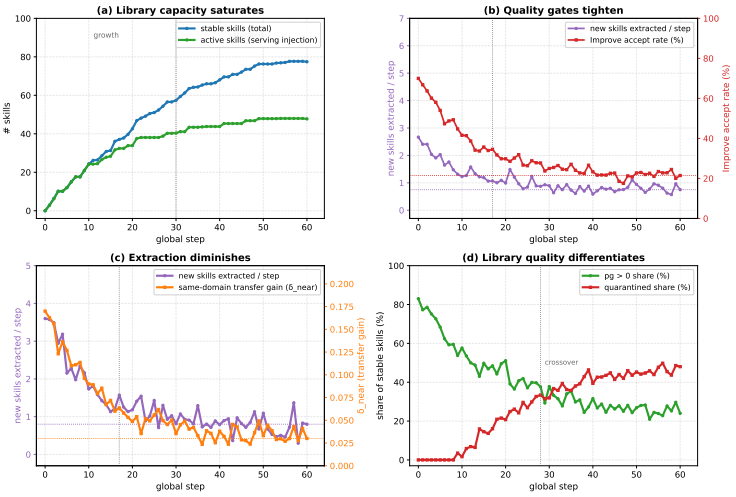

Figure 3: Evolution of the skill library.

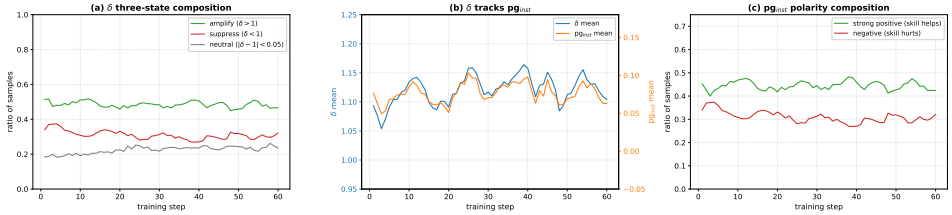

Figure 4: Instance-level causal advantage modulation during training.

6 Conclusion

This work identifies skill credit blindness as a central obstacle in skill-based reinforcement learning. When rollouts generated under different skill conditions share a group-relative baseline, GRPO cannot distinguish trajectory quality from the contribution of the injected skill. To address this problem, SCPO treats skills as explicit conditioning variables and separates the credit into behavior credit and skill credit. Behavior credit is estimated within matched problem-skill groups, while skill credit is estimated through same-problem counterfactual comparisons and incorporated into the update via causal advantage modulation. SCPO further aggregates these counterfactual skill effects across problems to guide the skill library as an evolving component of the RL loop. Experiments suggest that effective skill-based reinforcement learning requires not only learning and applying reusable skills, but also determining how each skill should influence policy updates and whether it should remain available for future tasks. A limitation of this work is that SCPO currently assumes that each rollout is conditioned on at most one skill and therefore cannot model interactions among multiple skills. Its paired utility estimates can also be noisy when rewards are sparse or when both conditions fail. Moreover, the present evaluation is limited to Qwen3-series models and mathematical reasoning tasks. Extending SCPO to compositional skill use, denser process-level feedback, and broader task domains is an important direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation and language polishing, and to help create or edit software code used during the project. We did not use generative AI tools to develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proofs, write proofs, propose or refine hypotheses, design research methodology or experiments, support qualitative or thematic data analysis, or interpret results. Generating synthetic datasets and cleaning or reformatting datasets are not applicable to this work. We reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personal data, or any privacy-sensitive information. All datasets used in our experiments are publicly available benchmark datasets, and we used them in accordance with their original licenses and intended purposes. We did not collect or release any new dataset that could raise concerns regarding consent, privacy, or data governance.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our work. The experimental setup, including model configurations, evaluation protocol, and prompt templates, is described in the main text and appendix. All datasets used in our experiments are publicly available benchmarks, and we report the evaluation metric, model scales, ablation settings, and the use of multiple random seeds to make the reported comparisons reproducible.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024. doi: 10.18653/v1/2024.acl-long.662.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv preprint*, 2022. URL https://arxiv.org/abs/2204.05862v1.

Zhiyuan Fan, Wenwei Jin, Feng Zhang, Bin Li, Yihong Dong, Yao Hu, and Jiawei Li. Evolving-rl: End-to-end optimization of experience-driven self-evolving capability within agents, 2026. URL https://arxiv.org/abs/2605.10663.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Yanzhou Li, Yiran Zhang, Xiaoyu Zhang, Xiaoxia Liu, and Yang Liu. Codeskill: Learning self-evolving skills for coding agents, 2026a. URL https://arxiv.org/abs/2605.25430.

Yu Li, Rui Miao, Zhengling Qi, and Tian Lan. Arise: Agent reasoning with intrinsic skill evolution in hierarchical reinforcement learning, 2026b. URL https://arxiv.org/abs/2603.16060.

Mathematical Association of America. 2025 american invitational mathematics examination (aime). https://maa.org/maa-invitational-competitions/, 2025.

Shuai Pan, Yixiang Liu, Jiaye Gao, Te Gao, Weiwen Liu, Jianghao Lin, Zhihui Fu, Jun Wang, Weinan Zhang, and Yong Yu. Skillmas: Skill co-evolution with llm-based multi-agent system, 2026. URL https://arxiv.org/abs/2605.09341.

Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. ART: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint*, 2023. URL https://arxiv.org/abs/2303.09014v1.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs, 2025. URL https://arxiv.org/abs/2504.13958.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language Models Can Teach Themselves to Use Tools. *arXiv preprint*, 2023. URL https://arxiv.org/abs/2302.04761v1.

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

Devan Shah, Owen Yang, Daniel Yang, Chongyi Zheng, and Benjamin Eysenbach. Upskill: Mutual information skill learning for structured response diversity in llms, 2026. URL https://arxiv.org/abs/2602.22296.

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

Junhao Shen, Teng Zhang, Xiaoyan Zhao, and Hong Cheng. Dynamic skill lifecycle management for agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2605.10923.

Yaorui Shi, Yuxin Chen, Zhengxi Lu, Yuchun Miao, Shugui Liu, Qi GU, Xunliang Cai, Xiang Wang, and An Zhang. Skill1: Unified evolution of skill-augmented agents via reinforcement learning, 2026. URL https://arxiv.org/abs/2605.06130.

Yash Vishe, Rohan Surana, Xunyi Jiang, Zihan Huang, Xintong Li, Nikki Lijing Kuang, Tong Yu, Ryan A. Rossi, Jingbo Shang, Julian McAuley, and Junda Wu. Skill-r1: Agent skill evolution via reinforcement learning, 2026. URL https://arxiv.org/abs/2605.09359.

Shouang Wei, Houcheng Min, Xinpeng Dong, Xin Lin, Sen Cui, Bo Jiang, Zhongxiang Dai, Kun Kuang, Guandong Xu, Fei Wu, and Min Zhang. Metaforge: A self-evolving multimodal agent that retrieves, adapts, and forges tools on demand, 2026. URL https://arxiv.org/abs/2606.01801.

Xiyang Wu, Zongxia Li, Guangyao Shi, Alexander Duffy, Tyler Marques, Matthew Lyle Olson, Tianyi Zhou, and Dinesh Manocha. Co-evolving llm decision and skill bank agents for long-horizon tasks, 2026. URL https://arxiv.org/abs/2604.20987.

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

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

Ruiyi Yang, Zechen Li, Hao Xue, Imran Razzak, and Flora D. Salim. Mage: Multi-agent self-evolution with co-evolutionary knowledge graphs, 2026. URL https://arxiv.org/abs/2605.10064.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv preprint*, 2022. URL https://arxiv.org/abs/2210.03629v3.

Jianxiang Yu, Jiapeng Zhu, Bochen Lin, Qier Cui, Zichen Ding, and Xiang Li. Skill is not one-size-fits-all: Model-aware skill alignment for llm agents, 2026. URL https://arxiv.org/abs/2605.30723.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *arXiv preprint*, 2025. URL https://arxiv.org/abs/2503.14476v2.

Mingda Zhang, Tiesunlong Shen, Haoran Luo, Wenjin Liu, Zikai Xiao, Erik Cambria, and Xiaoying Tang. Skillflow: Flow-driven recursive skill evolution for agentic orchestration, 2026. URL https://arxiv.org/abs/2605.14089.

A Appendix

A.1 Skill Lifecycle Management

Once skills are externalized, the skill library $\mathcal{S}$ becomes an external capability asset independent of model parameters. However, a static skill library can become misaligned as training progresses. Some skills may be useful early but become redundant after the model internalizes the corresponding reasoning pattern. Some may be effective only for their source problem and fail to transfer. Others may have overly broad applicability conditions and be invoked in incorrect contexts.

SCPO uses library-level paired gain as a unified utility signal to drive skill-library evolution. For each skill s, SCPO maintains online statistics

$$\mathcal{M}_s = \left(n_s^{\mathrm{tr}}, \bar{\Delta}_s^{\mathrm{tr}}, c_s, h_s\right), \tag{18}$$

where n_s^{tr} is the number of paired transfer evaluations, $\bar{\Delta}_s^{\mathrm{tr}}$ is the exponential moving average of cross-problem gain, c_s is the cumulative number of invocations, and h_s is the harmful invocation rate.

Each skill is assigned one of four stable states:

$$z_s \in \{\mathrm{Candidate}, \mathrm{Local}, \mathrm{Global}, \mathrm{Quarantined}\}. \tag{19}$$

Here, $\mathrm{Candidate}$ denotes a newly extracted skill with insufficient evidence; Local denotes a skill that is used only for its source task or nearby tasks; Global denotes a skill with cross-task transfer value; and $\mathrm{Quarantined}$ denotes a skill temporarily excluded from injection.

A.1.1 Skill Extraction

When successful and failed trajectories coexist for the same problem, SCPO extracts candidate skills from their contrast. We define the successful and failed trajectory sets as

$$\mathcal{Y}^+(x) = \left\{y_i \mid r(x, y_i) \geq \tau^+\right\}, \qquad \mathcal{Y}^-(x) = \left\{y_i \mid r(x, y_i) \leq \tau^-\right\}, \tag{20}$$

where τ^+ and τ^- are success and failure thresholds. The skill extractor compares the reasoning structures in $\mathcal{Y}^+(x)$ and $\mathcal{Y}^-(x)$, identifies reasoning patterns that may explain success, and produces a new skill

$$s = (\mathrm{name}, \mathrm{when}, \mathrm{how}, \mathrm{why}). \tag{21}$$

The field when specifies both applicability and exclusion conditions, how provides an executable reasoning template, and why explains the underlying rationale. Before entering active use, a newly extracted skill must pass validation and accumulate sufficient paired evidence before being promoted to a local or global skill.

A.1.2 Skill Selection

Given a problem x, SCPO retrieves candidate skills from the library for possible injection. To preserve clean attribution, at most one skill is active for each trajectory. If multiple skills are injected simultaneously, reward differences would mix the main effects and interactions of several skills, making pg_{inst} no longer attributable to a single skill.

Skill selection balances exploitation and exploration. For each candidate skill s, SCPO computes a composite score

$$\mathrm{Score}(x, s) = \lambda_{\mathrm{sim}}\mathrm{Sim}(x, s) + \lambda_{\mathrm{util}}U_s + \lambda_{\mathrm{dom}}D(x, s) + \lambda_{\mathrm{exp}}B_s. \tag{22}$$

Here, $\mathrm{Sim}(x, s)$ is the embedding similarity between the problem and the skill applicability condition, U_s is the historical utility estimate, $D(x, s)$ is a domain-matching bonus, and B_s is an exploration term:

$$B_s = \sqrt{\frac{\log\left(1 + \sum_{s' \in \mathcal{S}} c_{s'}\right)}{1 + c_s}}. \tag{23}$$

The exploration term is larger when a skill has limited evidence and decays as the skill is invoked more frequently. SCPO greedily selects the top-K skills by this score, and the model then chooses whether to inject one selected skill through an explicit skill-use decision.

A.1.3 Skill Evolution

The goal of skill evolution is not to simply add or delete skills, but to keep the library useful and transferable as the policy changes. During training, a skill may undergo three typical failure modes. It may remain useful but have an overly broad applicability condition; it may work only for its source task but fail to transfer; or it may consistently produce negative gains and contaminate future training. SCPO models skill evolution as an evidence-driven state update process.

For a skill s, SCPO collects two sets of problems from recent paired evaluations: problems on which the skill produces positive gain, and problems on which it produces negative gain:

$$\mathcal{P}_s = \{x \mid \Delta_s(x) > \epsilon_+\}, \qquad \mathcal{N}_s = \{x \mid \Delta_s(x) < -\epsilon_-\}. \tag{24}$$

The positive set $\mathcal{P}_s$ characterizes where the skill should be invoked, while the negative set $\mathcal{N}_s$ characterizes where it should be avoided. Based on these examples, SCPO generates a refined applicability condition $\widetilde{\text{when}}$. This step does not change the core reasoning procedure of the skill; it only revises the boundary of when the skill should be used. If the candidate condition is accepted, the skill is updated; otherwise, SCPO retains the original condition and continues collecting evidence. This prevents the library from oscillating under noise while also avoiding persistent misuse caused by overly coarse initial descriptions.

After local refinement, SCPO further evaluates whether a skill has cross-problem transfer value. A skill that is useful for its source problem should not necessarily become globally available. To distinguish local usefulness from transferable utility, SCPO uses a conservative lower-confidence estimate of cross-problem paired gain:

$$\mathrm{LCB}_s = \bar{\Delta}_s^{\mathrm{tr}} - \beta \cdot \frac{\hat{\sigma}_s^{\mathrm{tr}}}{\sqrt{n_s^{\mathrm{tr}}}}. \tag{25}$$

Here, $\bar{\Delta}_s^{\mathrm{tr}}$ is the mean paired transfer gain of skill s, $\hat{\sigma}_s^{\mathrm{tr}}$ is the standard deviation of the gain, n_s^{tr} is the number of transfer evaluations, and β controls the conservativeness of the estimate. Compared with directly using the mean gain, the lower-confidence bound promotes a skill only when it shows stable positive utility across multiple problems.

If the number of transfer evaluations is insufficient, i.e., $n_s^{\mathrm{tr}} < n_{\min}$, SCPO does not update the skill state. Otherwise, the state transition follows

$$z_s \leftarrow \begin{cases} \text{Local}, & z_s = \text{Candidate} \ \wedge \ \bar{\Delta}_s^{\mathrm{tr}} > \epsilon_{\mathrm{local}}, \\ \text{Global}, & z_s \in \{\text{Candidate}, \text{Local}\} \ \wedge \ \mathrm{LCB}_s > \epsilon_{\mathrm{global}} \ \wedge \ h_s < h_{\max}, \\ \text{Quarantined}, & \mathrm{LCB}_s < -\epsilon_{\mathrm{harm}} \ \vee \ h_s > h_{\max}, \\ z_s, & \text{otherwise}. \end{cases} \tag{26}$$

These rules correspond to three outcomes. A candidate skill with stable positive utility but insufficient transfer evidence is promoted to a local skill and used only for nearby problems. A skill

with a high lower-confidence gain and a low harmful invocation rate is promoted to a global skill. Conversely, a skill with negative conservative utility or frequent harmful invocations is quarantined and excluded from training-time injection.

Through this mechanism, SCPO turns skill-library evolution from heuristic maintenance into a closed loop. Refinement reduces mis-invocation, transfer evaluation determines generalizability, and state transitions control the visibility of skills during training. The library therefore does not accumulate unverified prompts indefinitely, nor does it prematurely discard locally useful skills. Instead, it gradually becomes a retrievable, calibratable, and transferable external capability system.

A.1.4 SKILL EXAMPLE

Table 3: Representative stable skills extracted by SCPO

Name	**When**	**Why**	**How**
State transition with odd/even count	Recursive or iterative processes involving state transitions with alternating parity; not applicable to algebra problems.	A state-transition model captures the sequence structure, and parity conditions isolate the target probability.	Define a state tracking the current matching progress, build the state-transition equation with a parity counter, and recurse toward the target probability.
Solve system of equations with constraints	A quadratic equation coupled with a linear equation, under additional constraints.	Substitution-based elimination turns the multi-variable problem into a single-variable one that is easier to solve.	Substitute the linear equation into the quadratic one, reduce it to a single quadratic, then verify the constraints on the solutions.
Identify equal elements in a set	A set whose elements may coincide and whose equality cases must be enumerated; not applicable to geometry problems.	Explicit equality conditions narrow the range of admissible combinations.	Analyze the possible values of each element, determine when two or three elements coincide, and prune redundant cases.
Identify the key equation from conditions	A stated condition that uniquely determines one core equation directly linking the target quantity; not applicable to geometry problems.	Translating stated conditions into an algebraic equation is the core strategy for this class of problems.	Convert the condition into an algebraic equation via substitution or equivalent transformation, then solve for the parameter.

A.2 PROMPT TEMPLATES

This section documents the prompts used in training. All rollouts of a pair share the same system prompt and the same problem statement; the skill and no-skill conditions differ only in the skill cards presented before the problem. The prompt may list up to K candidate skill cards (K=2 in our runs); the model activates at most one skill per trajectory through the selection tag, preserving single-skill attribution. Prompts are translated from the original Chinese for presentation.

System prompt (shared by all rollouts).

```
You are an AI assistant skilled at mathematical reasoning.

Answer format requirements:
```

```
1. First select a problem-solving skill inside the <skill> tag
   (if a list of available skills is provided with the problem);
   fill in "none" if none applies.
2. Carry out detailed reasoning inside the <think> tag
   (clear logic, complete steps).
3. Put the final answer inside the <answer> tag
   (answer only, no derivation).
```

User message, no-skill condition (control).

```
Solve the following problem:
{problem}
```

User message, skill-injected condition (treatment).

```
[Skill: {name}]
Applicable when: {when}
Method: {how}
Rationale: {why}
Confidence: {1-3 stars} | Mark the injected skill inside
<skill>{name}</skill>, or write <skill>none</skill> if it does not
apply.

Solve the following problem:
{problem}
```

A fully instantiated example, using the skill card named `solve_system_of_equations_with_constraints`:

```
[Skill: solve_system_of_equations_with_constraints]
Applicable when: a quadratic equation is coupled with a linear
equation under additional constraints.
Method: substitute the linear equation into the quadratic one,
reduce it to a single quadratic, then verify the constraints on
the solutions.
Rationale: substitution-based elimination turns the multi-variable
problem into a single-variable one that is easier to solve.
Confidence: *** | Mark the injected skill inside
<skill>solve_system_of_equations_with_constraints</skill>, or write
<skill>none</skill> if it does not apply.

Solve the following problem: {problem}
```

The confidence level reflects the skill state and its accumulated injection lift: a quarantined or unstable skill is never injected, a stable skill receives two stars, and a stable skill whose lift exceeds a calibrated threshold receives three stars. The skill-selection tag emitted by the model is parsed to attribute each rollout to its skill condition, which defines the grouping key (x, s) used by skill-conditioned normalization and the paired skill gain estimate.