Skill-Coupled Policy Optimization with Calibrated Group-Wise Advantage Estimation

Peng Wang[1] **Zhendong Chu**[1] **Chengshuai Shi**[2]
Matthias Poloczek[3*] **Cong Shen**[1] **Jing Yang**[1]
[1]University of Virginia [2]Princeton University [3]Databricks
{pw7nc,zc9uy,cong,yangjing}@virginia.edu
cs1083@princeton.edu matthias.poloczek@databricks.com

Abstract

Reinforcement learning with verifiable rewards (RLVR) has become a standard paradigm for post-training large language models on reasoning tasks. To avoid training explicit critics, methods such as GRPO estimate advantages from sparse on-policy rollouts, making baseline accuracy critical for effective optimization. In this work, we argue that commonly used baselines, whether relying on prompt-local averages or shrinking toward a single global mean, are statistically suboptimal for reasoning domains. By enforcing a uniform prior, these methods overlook structured heterogeneity in task difficulty and correlations among related problems, leading to biased and noisy advantage estimates. To address this, we propose *Skill-Coupled Policy Optimization* (SCPO), a structured baseline estimation framework that couples value estimation across tasks sharing similar reasoning skills. Instead of a global baseline, SCPO performs shrinkage toward skill-coupled group baselines, thereby aligning the baseline with the local difficulty landscape. To further stabilize these baselines with limited on-policy data, SCPO incorporates a lightweight history tracker. Experiments across diverse models and tasks show that SCPO consistently improves training stability and performance over both prompt-level and global shrinkage baselines. Code is available at `https://github.com/peter-peng-w/SCPO`.

1 Introduction

Reinforcement-learning (RL)-based post-training has become a standard approach for improving the reasoning capability of large language models (LLMs) in mathematical [41, 19], coding [25, 29], and scientific domains [10, 38]. While effective, the pioneering *actor-critic* paradigm introduces substantial training complexity and GPU memory overhead, and critic training is often brittle under distribution shift, making it a key engineering bottleneck in large-scale deployment [2, 3]. Recent reasoning-centric methods increasingly adopt *critic-free* reinforcement learning with verifiable rewards (RLVR), which avoids training explicit critics by estimating advantages directly from on-policy rollouts and has proven effective at scale [11]. However, the critic-free nature of RLVR makes optimization highly sensitive to baseline accuracy: with only a few rollouts per prompt, *how can we estimate a reliable baseline that avoids high-variance advantages and unstable policy updates?*

A standard practice in critic-free RLVR is to use the prompt-level mean accuracy computed from M on-policy rollouts as the baseline [11]; however, this estimator becomes highly noisy when M is small [21]. The resulting noise directly propagates into the advantage estimates, increasing gradient variance and destabilizing learning [13, 51, 49]. Recent work has made progress by improving

*This research does not relate to Matthias' work at Amazon and Databricks.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

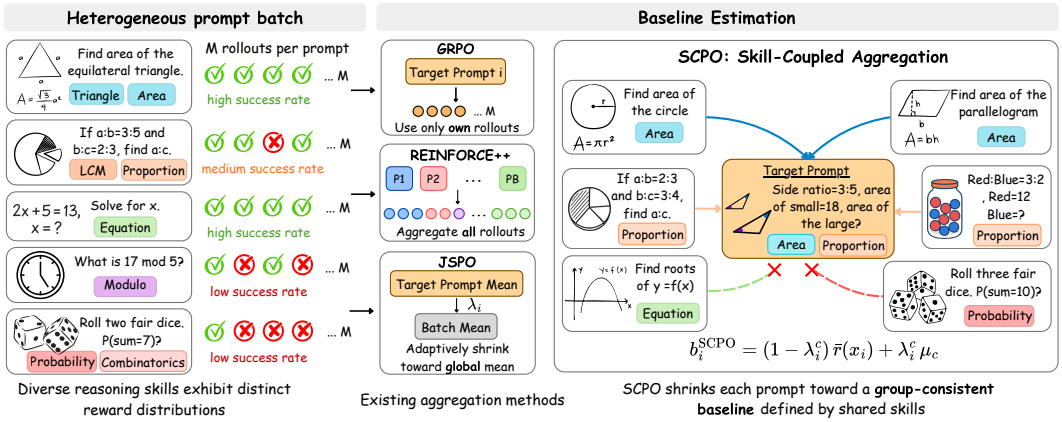


Figure 1: **Overview of SCPO. (Left)** A training batch is essentially heterogeneous as prompts span diverse reasoning skills with reward distributions that differ substantially across skills. **(Middle)** Existing critic-free baselines aggregate over different scopes, for example, GRPO uses only the target prompt's own rollouts, REINFORCE++ averages over the entire batch, and JSPO adaptively shrinks the prompt mean toward the global batch mean – but all treat aggregated prompts as exchangeable. **(Right)** SCPO instead shrinks each prompt's on-policy mean $\bar{r}(x_i)$ toward a group-consistent target μ_c defined by shared skills, yielding the baseline $b_i^{\mathrm{SCPO}} = (1-\lambda_i^c)\,\bar{r}(x_i) + \lambda_i^c\,\mu_c$.

the baseline through *cross-prompt aggregation*, pooling reward statistics across prompts to reduce finite-sample variance. REINFORCE++ [21] advocates global advantage normalization by computing the baseline as the mean accuracy of BM rollouts from a batch of B prompts to stabilize training. JSPO [51] proposes a James–Stein shrinkage baseline defined as a weighted average of the prompt-level mean accuracy (as in GRPO) and a batch-level mean accuracy (similar to REINFORCE++). The shrinkage weight is jointly determined by the variance of the M rollouts of the target prompt and the variance of the B prompt-level mean accuracies across the batch, yielding a substantially lower-variance baseline estimate.

However, existing cross-prompt aggregation methods implicitly treat aggregated prompts as statistically *homogeneous*. In reasoning datasets, prompts often involve different skills and difficulty levels. For example, as illustrated in Figure 1, prompts involving simple arithmetic, multi-step geometric reasoning, or combinatorics may have substantially different success probabilities under the same policy. Recent work [6, 27, 54, 23] also characterizes reasoning tasks in terms of shared constituent skills and their relationships, providing a structured perspective on task organization in language models. As a result, cross-prompt aggregation that ignores these differences can produce a batch mean that poorly matches individual prompts' expected rewards. Whether used directly as a baseline or as a target for global shrinkage, this mean can introduce structured bias into prompt-level baseline estimates, distorting advantage estimates and limiting the benefits of variance reduction.

We empirically substantiate this heterogeneity using skill annotations derived from a metacognitive skill taxonomy for mathematical domain [9]. Specifically, on the MATH dataset [20], we sample 8 responses per question using Qwen2.5-Math-7B-Instruct and compute prompt-level mean accuracy. As depicted in Figure 5, aggregating by skills reveals pronounced stratification, with some skills achieving consistently high accuracy and others substantially lower. This indicates that baseline shrinkage under a homogeneity assumption is statistically mis-specified given latent group heterogeneity.

Motivated by these observations, we propose ***S**kill-**C**oupled **P**olicy **O**ptimization* (SCPO), a group-aware value estimation framework for critic-free RLVR. As illustrated in Figure 1, the central idea is to replace global batch-level aggregation with *within-group shrinkage*. To account for this skill-conditioned heterogeneity, we organize prompts into groups around shared reasoning skills, using skill annotations or semantic similarity to identify related tasks. Instead of shrinking each prompt's on-policy mean reward toward a single batch-level mean, SCPO shrinks toward a *skill-level* target estimated from a group of related prompts. Restricting shrinkage to prompts that are likely to have similar expected rewards can reduce baseline estimation error by sharing reward information while limiting cross-group mismatches. James–Stein shrinkage [43, 22, 51] provides one principled instantiation of this within-group estimator, and SCPO provides a general framework for incorporating

such shrinkage in a way that respects dataset heterogeneity. Conceptually, SCPO reframes baseline construction in critic-free RLVR as a structured value-estimation problem.

While within-group shrinkage can reduce target mismatch, reliable estimation of group-level targets remains challenging under a limited rollout budget. Each group contributes only a subset of the current batch's rollouts, so its target estimate can remain noisy. To stabilize these targets without training a critic, we introduce a lightweight history-based *group value tracker* that maintains recent prompt-level reward statistics within each group and aggregates them across training iterations. This increases the effective sample size for each group target, incurring negligible computational overhead. From an empirical Bayes perspective, the tracker estimates a prior mean for each skill group, paralleling the use of a global running prior in concurrent EBPO [12]. These histories also provide difficulty estimates before new rollouts are generated, naturally supporting an optional curriculum within skill groups without additional rollouts.

In summary, we advance critic-free RLVR through accurate, heterogeneity-aware baseline estimation. Our main contributions are threefold.

- We empirically demonstrate substantial skill-conditioned differences in expected rewards in mathematical reasoning data, motivating skill-aware baseline estimation.
- We propose SCPO, a group-aware framework that performs within-group shrinkage to align baseline estimation with shared skill structure, with James–Stein shrinkage as a principled instantiation.
- We introduce history-based group value tracking to stabilize target estimation and reuse its prompt-level reward statistics for an optional within-group curriculum without additional rollouts.

Experiments across models and reasoning tasks demonstrate SCPO's effectiveness, while ablations assess the contributions of grouping, history tracking, and curriculum learning.

2 Preliminaries

Reinforcement Learning with Verifiable Rewards. Let π_θ denote a language model parameterized by θ. Given a prompt $x \sim \mathcal{D}$, the model samples a response $y \sim \pi_\theta(\cdot \mid x)$ and receives a scalar reward $r(x, y)$. In RLVR, r is a deterministic verifier-based function of (x, y), often binary or sparse. The objective is to maximize the expected reward $J(\theta) := \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}[r(x, y)]$, which policy-gradient methods optimize via $\nabla_\theta J(\theta) = \mathbb{E}_{x,y}[r(x, y)\, \nabla_\theta \log \pi_\theta(y \mid x)]$. A standard variance-reduction technique introduces a prompt-dependent baseline $b(x)$ that is independent of the sampled reward given x, yielding the unbiased policy-gradient estimator $g(x, y; \theta) := (r(x, y) - b(x))\, \nabla_\theta \log \pi_\theta(y \mid x)$. We focus on critic-free settings where $b(x)$ is estimated from sampled rollouts rather than learned by an explicit value network, as in actor–critic methods such as PPO [40]. By avoiding an explicit critic, critic-free RLVR reduces training complexity and GPU memory overhead and has therefore become a common choice for LLM post-training.

Across different critic-free methods [41, 45, 14, 31], the central design axis is the baseline estimator $b(x)$ and its induced advantage normalization. GRPO relies on prompt-local mean estimation, REINFORCE++ stabilizes optimization via global batch normalization, and JSPO further reduces baseline estimation error via adaptive shrinkage of prompt-level estimates toward a batch-level mean, with a leave-one-out construction to preserve unbiased policy-gradient estimates. In contrast, our method builds on this perspective and introduces group-aware shrinkage and group-level tracking to better match the latent structure of post-training tasks.

Group Relative Policy Optimization (GRPO). Consider a batch of n prompts $\{x_i\}_{i=1}^n$. For each prompt x_i, GRPO [41] samples m on-policy rollouts $\{y_i^j\}_{j=1}^m \sim \pi_\theta(\cdot \mid x_i)$ with rewards $\{r_i^j\}_{j=1}^m$. GRPO avoids training an explicit critic by constructing per-sample advantages using prompt-local statistics together with Z-score normalization

$$A_{i,j}^{\mathrm{GRPO}} = \frac{r_i^j - \mu_i}{\sigma_i + \varepsilon}, \quad \mu_i = \frac{1}{m}\sum_{j=1}^m r_i^j, \quad \sigma_i = \sqrt{\frac{1}{m-1}\sum_{j=1}^m \Big(r_i^j - \mu_i\Big)^2}, \tag{1}$$

where $\varepsilon > 0$ avoids division by zero. Although simple and critic-free, prompt-level normalization can be noisy when m is small. Moreover, each reward contributes to the mean and standard deviation used to normalize it, which can introduce finite-sample conditional bias in advantage estimates [21, 49],

potentially destabilizing training. The corresponding optimization objective can be expressed as

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{x_i \sim \mathcal{D} \\ \{y_i^j\}_{j=1}^m \sim \pi_{\theta_{\text{old}}}}} \left[\frac{1}{m} \sum_{j=1}^{m} \frac{1}{|y_i^j|} \sum_{t=1}^{|y_i^j|} \min\Big(s_{i,j}^t(\theta) A_{i,j}^{\text{GRPO}}, \text{clip}\big(s_{i,j}^t(\theta), 1-\epsilon, 1+\epsilon\big) A_{i,j}^{\text{GRPO}} \Big) \right] \tag{2}$$

where θ denotes the current policy parameters and θ_{old} denotes the policy used to collect the rollouts, and $s_{i,j}^t(\theta) = \frac{\pi_\theta(y_i^{j,t} | x_i, y_i^{j,<t})}{\pi_{\theta_{\text{old}}}(y_i^{j,t} | x_i, y_i^{j,<t})}$ represents the importance sampling ratio.

Variance Reduction via Global Normalization (REINFORCE++). REINFORCE++ [21] stabilizes critic-free optimization through *global* batch normalization and supports $m \geq 1$ rollouts per prompt. In a sequence-level formulation without KL reward shaping, its basic variant computes

$$A_{i,j}^{\text{GN}} = \frac{r_i^j - \mu_{\text{batch}}}{\sigma_{\text{batch}} + \varepsilon}, \qquad \mu_{\text{batch}} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} r_i^j, \tag{3}$$

where σ_{batch} is the standard deviation of all nm rollout rewards in the batch. For $m > 1$, the optional local-baseline variant treats the rollouts of each prompt as a group, subtracts their mean reward, and then applies global normalization

$$A'_{i,j} = r_i^j - \mu_i, \qquad A_{i,j}^{\text{GN}} = \frac{A'_{i,j} - \mu'_{\text{batch}}}{\sigma'_{\text{batch}} + \varepsilon}, \tag{4}$$

where μ_i is the prompt mean in Equation (1), and μ'_{batch} and σ'_{batch} are the mean and standard deviation of the centered rewards $\{A'_{i,j}\}_{i,j}$ across the full batch.

Shrinkage Baselines via Joint Mean Estimation (JSPO). JSPO [51] studies baseline estimation as a joint mean-estimation problem across prompts. Let $\hat{\mu}_i = \frac{1}{m} \sum_{j=1}^{m} r_i^j$ denote the prompt-level mean reward and $\hat{\bar{\mu}} = \frac{1}{n} \sum_{i=1}^{n} \hat{\mu}_i$ the global batch mean. A James–Stein-style shrinkage estimator [22, 43] interpolates between them as $b_i^{\text{JS}} = (1-\lambda)\, \hat{\mu}_i + \lambda\, \hat{\bar{\mu}}$, where the shrinkage coefficient $\lambda \in [0, 1]$ trades bias in prompt-value estimation against variance reduction.

To avoid bias from reusing the evaluated response in its own baseline, JSPO applies the standard leave-one-out principle at both the prompt and batch levels, following RLOO [1]. For $m > 1$ and $n > 1$, define the prompt-level LOO mean $\hat{\mu}_i^{-j} = \frac{1}{m-1} \sum_{k \neq j} r_i^k$ and the batch-level LOO mean $\hat{\bar{\mu}}_{-i} = \frac{1}{n-1} \sum_{\ell \neq i} \hat{\mu}_\ell$. These yield the shrinkage baseline $b_{i,j}^{\text{JS}} = (1 - \hat{\lambda}_i)\, \hat{\mu}_i^{-j} + \hat{\lambda}_i\, \hat{\bar{\mu}}_{-i}$.

The shrinkage weight admits the closed-form estimate $\hat{\lambda}_i = \frac{n-1}{n} \cdot \frac{\hat{v}_{-i}}{\hat{v}_{-i} + \hat{s}_{-i}}$, where $\hat{v}_{-i}$ measures within-prompt estimation noise and $\hat{s}_{-i}$ measures dispersion across prompt-level means. Both statistics are computed using only rollouts from prompts other than x_i. Greater estimation noise relative to cross-prompt dispersion favors stronger shrinkage toward the batch-level target. Under independent rollout sampling, this construction makes the baseline independent of the held-out response y_i^j conditional on x_i, preserving unbiased estimation of the underlying policy gradient. This baseline is a drop-in replacement for the prompt-mean baseline with negligible overhead.

3 Methodology

We propose **S**kill-**C**oupled **P**olicy **O**ptimization (SCPO), a group-aware baseline estimation framework for critic-free RLVR. SCPO constructs baselines that stabilize advantage estimation by performing *within-group shrinkage*, pooling reward statistics among prompts that are semantically or skill-related, and further stabilizes group targets using a short recent-history buffer at the group level. This section formalizes the grouping and sampling pipeline, presents the SCPO baseline estimator, and introduces history-based group target tracking.

3.1 Prompt Grouping by Semantics and Skills

As demonstrated in Figure 5, LLM post-training corpora are generally heterogeneous, with prompts varying in topic, difficulty, and thus require different reasoning skills. Global aggregation thus can

over-pool dissimilar prompts and induce systematic mismatch in baseline estimation. SCPO therefore organizes prompts into C groups and estimates shrinkage targets *within* each group.

Group construction. We consider two complementary grouping mechanisms. First, **semantic groups** are obtained by embedding prompts (or prompt metadata) via off-the-shelf embedding models, and applying a clustering algorithm such as k-means. Second, **skill groups** are formed from discrete skill labels associated with each prompt, for example labels predicted by a tagger or derived from a pre-defined skill taxonomy [9]. Once the pre-processing finish, we receive a fixed partition $\mathcal{C} = \{c\}_{c=1}^{C}$, where $c(x)$ indicates the group indices of prompt x. Without loss of generality, each $c(x) = \{c_k\}_{k=1}^{K}$, denoting that each prompt can be associate with more than one group.

Group-balanced batch sampling. Both semantic and skill-based groups typically follow a long-tail distribution, with a small number of frequent groups covering most of the corpus. To provide stable updates across frequent and rare groups, we construct each training batch by first sampling a subset of groups and then sampling prompts within each selected group. Following prior work [18], we sample groups with probability proportional to their group size to prevent over-sampling the long tail. At each iteration, we sample G groups and draw P prompts per group, yielding a batch of $B = G \cdot P$ prompts. For each prompt, we generate M on-policy rollouts, resulting in $B \cdot M$ reward-labeled samples per iteration, which we use to estimate prompt means and update group targets.

3.2 Within-Group Shrinkage Baselines

We now describe the SCPO baseline estimator. For a prompt x_i sampled from group c, let $y_i^j \sim \pi_\theta(\cdot \mid x_i)$ be its j-th on-policy rollout with reward r_i^j, for $j = 1, \ldots, M$. Its mean reward $\bar{r}(x_i) = \frac{1}{M}\sum_{j=1}^{M} r_i^j$ corresponds to the prompt-level mean $\hat{\mu}_i$ in the JSPO formulation. Let μ_c denote the shrinkage target for group c (Section 3.3). SCPO combines the two estimates as

$$b_i^{\mathrm{SCPO}} = (1 - \hat{\lambda}_i^c)\, \bar{r}(x_i) + \hat{\lambda}_i^c\, \mu_c, \tag{5}$$

where $\hat{\lambda}_i^c \in [0, 1]$ is the estimated shrinkage weight. The advantage of rollout j is $A_{i,j} = r_i^j - b_{i,j}^{\mathrm{SCPO}}$, with $b_{i,j}^{\mathrm{SCPO}} = b_i^{\mathrm{SCPO}}$ when using the full prompt mean.

Leave-one-out stabilization. Reusing a rollout's reward in the means that form its baseline can bias the policy-gradient estimate. We thus apply the leave-one-out (LOO) principle [1, 24] to both means to avoid this direct reuse. For $M > 1$, the prompt-level LOO mean for rollout j is

$$\bar{r}^{-j}(x_i) = \frac{1}{M-1} \sum_{k \neq j} r_i^k. \tag{6}$$

We likewise form a LOO group target $\mu_{c,-i}$ using the same window as μ_c and excluding the current-batch rollouts of x_i. The resulting baseline is $b_{i,j}^{\mathrm{SCPO}} = (1 - \hat{\lambda}_i^c)\, \bar{r}^{-j}(x_i) + \hat{\lambda}_i^c\, \mu_{c,-i}$.

Shrinkage operator and James–Stein instantiation. Equation (5) gives a general form for within-group shrinkage. With few rollouts per prompt, sharing reward information across related prompts can reduce estimation variance. Shrinking toward a shared target can also introduce bias when expected rewards vary within a group. James–Stein shrinkage motivates a weight that adapts to this bias–variance trade off [22, 43, 51]. For $M > 1$ and $P > 1$, our plug-in coefficient is

$$\hat{\lambda}_i^c = \frac{\hat{\sigma}_i^2}{\hat{\sigma}_i^2 + \hat{S}_c^2}. \tag{7}$$

We set $\hat{\lambda}_i^c = 0$ when both variance estimates vanish. Let $\mathcal{B}_c$ denote the P prompts sampled from group c in the current batch, with mean $\bar{r}_c^{\mathrm{cur}} = \frac{1}{P}\sum_{x \in \mathcal{B}_c} \bar{r}(x)$. The two variance estimates are

$$\hat{\sigma}_i^2 = \frac{1}{M(M-1)} \sum_{j=1}^{M} \big(r_i^j - \bar{r}(x_i)\big)^2, \qquad \hat{S}_c^2 = \frac{1}{P-1} \sum_{x \in \mathcal{B}_c} \big(\bar{r}(x) - \bar{r}_c^{\mathrm{cur}}\big)^2.$$

The term $\hat{\sigma}_i^2$ estimates the sampling variance of $\bar{r}(x_i)$ by dividing the rollout reward sample variance by M. The term $\hat{S}_c^2$ measures the sample variance of all P observed prompt means in $\mathcal{B}_c$. Greater sampling noise relative to within-group dispersion places more weight on the group target.

3.3 History-Based Group Targets

Within-group shrinkage reduces mismatch relative to global aggregation, but group targets can remain noisy when P and M are small. SCPO therefore maintains a lightweight *group value tracker* that aggregates recent group-level reward statistics across iterations to stabilize μ_c.

On-policy group statistic. At training step t, we compute an on-policy reward statistic for each group c using the rollouts generated in the current batch. Let $\mathcal{B}_t$ denote the set of prompts sampled at step t, and recall that a prompt x may be associated with multiple groups $c(x) = \{c_k\}_{k=1}^{K}$. For each group c, we collect all rollouts whose prompts belong to c and compute the group mean reward

$$\hat{R}_{c,t} := \frac{1}{n_{c,t}} \sum_{\substack{x \in \mathcal{B}_t \\ c \in c(x)}} \sum_{j=1}^{M} r^{(j)}(x), \quad n_{c,t} := \sum_{\substack{x \in \mathcal{B}_t \\ c \in c(x)}} M, \tag{8}$$

where $n_{c,t}$ is the number of rollouts contributing to group c at step t.

Sliding-window target. We maintain, for each group c, a buffer that stores recent prompt-rollout tuples whose group membership includes c and whose generation time lies within the most recent W_t training steps. The group target is computed by aggregating reward statistics over this window

$$\mu_c(t) = \frac{\sum_{t' \in \mathcal{W}(t)} n_{c,t'} \, \hat{R}_{c,t'}}{\sum_{t' \in \mathcal{W}(t)} n_{c,t'}}, \; \mathcal{W}(t) := \{t - W_t + 1, \ldots, t\}, \tag{9}$$

Weighting each step by $n_{c,t'}$ gives each retained rollout equal weight. The window limits how long rewards from earlier policies can affect the target. Appendix C analyzes how target variance and policy drift affect shrinkage with a history-only target. Future work could compare this window with exponential moving averages [4] and policy-adaptive forgetting [45] for estimating group targets. The same rollout history supports an optional curriculum that selects a difficulty bin within each sampled skill group using per-prompt accuracy estimates.

3.4 Training Objective and Algorithm

SCPO modifies only the baseline and advantage estimation. Given per-sample advantages defined in Section 3.2, we optimize π_θ with a standard on-policy policy-gradient surrogate as used in GRPO-style RLVR training. Following Dr.GRPO [30], we omit the standard deviation in advantage normalization to avoid introducing problem-level biases. Furthermore, we set the reference-policy KL coefficient to zero, following prior work such as DAPO [50] and CISPO [5]. Algorithm 1 summarizes the full training procedure, including group target updates and SCPO advantage estimation.

4 Experiments

4.1 Experimental Setup

Models & Benchmarks. We conduct our experiments on Qwen2.5-Math-1.5B and Qwen2.5-Math-7B [48] to assess different baseline estimation methods' performance on mathematical reasoning tasks. Following DeepMath [19], we incorporate the following six benchmarks: AIME 2024 [34], AIME 2025 [34], AMC 23 [35], MATH-500 [20], Minerva [26], and OlympiadBench [15].

Evaluation Setup. As the default maximum context length for Qwen2.5-Math series models is 4,096, we set the maximum generation length to 3,072 tokens, and the maximum prompt length to 1,024 tokens, unless otherwise specified. All examples are retained, with overlong prompts being truncated. For the main experiments, we follow DeepSeek-R1 [11] and adopt the *Pass@k* evaluation protocol [7], reporting the average *Pass@1* with non-zero temperature sampling. Consistent with Yan et al. [46] and Lv et al. [33], we compute avg@32 for AIME 24, AIME 25, and AMC (avg@1 for others), using a temperature of 0.6 and a top-p value of 0.95 for the accuracy calculation.

Table 1: Accuracy (%) on six math benchmarks. Subscripts give standard deviations across three seeds (evaluation seeds for fixed checkpoints and training seeds for post-training runs). Avg. is the per-seed macro-average. The best and second-best post-training scores in each block are **bolded** and underlined, respectively.

Method	AIME 24	AIME 25	AMC 23	MATH-500	Minerva	Olympiad	**Avg.**
			Qwen2.5-Math-1.5B				
Base	7.7 $_{0.7}$	3.8 $_{0.3}$	31.8 $_{1.2}$	34.4 $_{2.1}$	12.0 $_{1.0}$	22.5 $_{1.1}$	18.73 $_{0.67}$
GRPO [41]	15.5 $_{0.4}$	11.0 $_{1.1}$	54.3 $_{1.4}$	**76.1** $_{0.4}$	35.7 $_{0.8}$	38.3 $_{0.4}$	38.49 $_{0.36}$
REINFORCE++ [21]	14.7 $_{0.4}$	10.6 $_{1.2}$	55.2 $_{1.0}$	74.5 $_{0.8}$	33.1 $_{0.3}$	37.4 $_{0.6}$	37.62 $_{0.45}$
JSPO [51]	15.3 $_{0.3}$	11.9 $_{1.0}$	56.3 $_{0.7}$	75.9 $_{0.8}$	35.4 $_{0.3}$	38.7 $_{0.9}$	38.92 $_{0.32}$
SCPO (ours)	**16.8** $_{0.9}$	**12.6** $_{1.3}$	**58.3** $_{0.5}$	76.0 $_{0.4}$	**36.4** $_{1.0}$	**39.9** $_{0.6}$	**39.98** $_{0.60}$
Instruct	10.7 $_{0.6}$	9.4 $_{0.9}$	53.9 $_{0.5}$	75.4 $_{0.3}$	33.7 $_{0.8}$	40.2 $_{0.3}$	37.25 $_{0.15}$
Oat-Zero [30]	16.3 $_{0.3}$	10.7 $_{0.5}$	55.0 $_{0.5}$	74.6 $_{0.5}$	35.3 $_{1.2}$	37.7 $_{0.3}$	38.25 $_{0.23}$
			Qwen2.5-Math-7B				
Base	11.5 $_{1.0}$	5.2 $_{0.4}$	36.1 $_{0.7}$	47.7 $_{3.2}$	15.1 $_{1.5}$	15.6 $_{1.9}$	21.88 $_{1.14}$
GRPO [41]	29.2 $_{1.2}$	13.8 $_{0.4}$	60.6 $_{0.8}$	80.7 $_{0.5}$	41.2 $_{0.6}$	44.5 $_{0.9}$	45.02 $_{0.64}$
REINFORCE++ [21]	27.4 $_{2.2}$	12.5 $_{1.4}$	58.8 $_{0.6}$	79.6 $_{0.4}$	40.3 $_{1.2}$	42.7 $_{0.8}$	43.55 $_{0.83}$
JSPO [51]	26.9 $_{2.6}$	11.8 $_{0.4}$	**63.8** $_{2.0}$	**81.5** $_{0.9}$	**42.7** $_{0.5}$	43.9 $_{1.0}$	45.10 $_{0.45}$
SCPO (ours)	**31.8** $_{1.1}$	**15.6** $_{0.8}$	63.2 $_{0.6}$	81.0 $_{0.4}$	41.8 $_{0.9}$	**45.2** $_{0.5}$	**46.43** $_{0.52}$
Instruct	11.1 $_{0.3}$	10.0 $_{1.0}$	57.8 $_{0.6}$	83.4 $_{0.6}$	41.3 $_{0.6}$	41.9 $_{0.5}$	40.92 $_{0.17}$
Oat-Zero [30]	30.8 $_{0.6}$	11.6 $_{0.3}$	65.2 $_{0.4}$	79.7 $_{0.5}$	40.5 $_{0.3}$	43.2 $_{0.1}$	45.17 $_{0.26}$

Implementation Details. We post-train Qwen2.5-Math-1.5B and Qwen2.5-Math-7B on Levels 3–5 of MATH [20]. Each iteration samples $G = 16$ groups, $P = 8$ prompts per group, and $M = 8$ on-policy rollouts per prompt, yielding a total batch size of $B = GP = 128$. Group-wise skill labels are obtained by prompting GPT-5-mini [42] with the metacognitive skill taxonomy of Didolkar et al. [9], returning up to five skills per question. Full optimizer settings, training step counts, and the sliding-window configuration are in Appendix E.1.

4.2 Main Results

Table 1 compares SCPO with GRPO, REINFORCE++, and JSPO on Qwen2.5-Math-1.5B and Qwen2.5-Math-7B. *SCPO achieves the highest macro-average accuracy at both scales* over three training seeds. At 7B, it reaches $(46.43 \pm 0.52)\%$, exceeding GRPO and JSPO by 1.41 and 1.33 percentage points, respectively. At 1.5B, it reaches $(39.98 \pm 0.60)\%$ with similar gains. REINFORCE++, which normalizes advantages across the batch, ranks last among the compared methods, suggesting that indiscriminate cross-prompt aggregation can be suboptimal. JSPO slightly improves on GRPO by adaptively combining noisy prompt-local estimates with a batch-level target. SCPO further restricts this aggregation to groups of prompts with shared reasoning skills.

SCPO's gains extend to the hardest evaluation benchmarks, as measured by base-model accuracy. Although all compared methods are post-trained on MATH, SCPO leads on AIME 2024 and AIME 2025, the two benchmarks with the lowest base-model accuracies at both scales. On AMC 23, SCPO leads at 1.5B and ranks second at 7B among the post-trained methods. Recent analyses [49] show that, when a prompt's rollouts contain both binary reward outcomes, mean-centered group-relative advantages can give too little credit to correct responses on hard prompts and too weak a penalty to incorrect responses on easy prompts. By aggregating within skill groups, SCPO can stabilize the cross-prompt baseline and mitigate the effects of heterogeneous prompt mixing, which may contribute to stronger performance on harder benchmarks.

4.3 Training Dynamics

Figures 2 and 3 track evaluation performance over training for three representative benchmarks, AIME 2024, AMC 23, and MATH-500, under different critic-free baseline estimators. In the runs shown, SCPO finishes ahead of GRPO and JSPO on AIME 2024 and AMC 23 at both model scales, while remaining competitive on MATH-500.

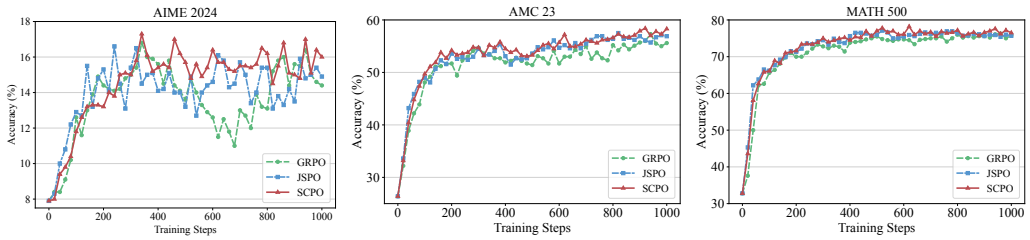


Figure 2: Evaluation accuracy over training for *Qwen2.5-Math-1.5B*. Each curve shows a single training run.

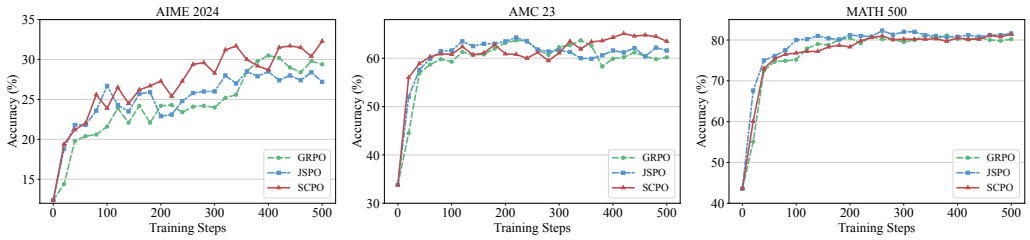


Figure 3: Evaluation accuracy over training for *Qwen2.5-Math-7B*. Each curve shows a single training run.

Beyond final performance, SCPO shows *smaller average fluctuations between successive evaluations* than GRPO and JSPO on AIME 2024 and AMC 23 in the 1.5B runs. This pattern is consistent with SCPO's group-aware baseline estimation. Shrinking toward group-consistent targets can reduce the mismatch from aggregating prompts with different expected rewards, while the short-horizon tracker pools more observations for each target and limits staleness. Together, these choices can help stabilize the baseline signal during training.

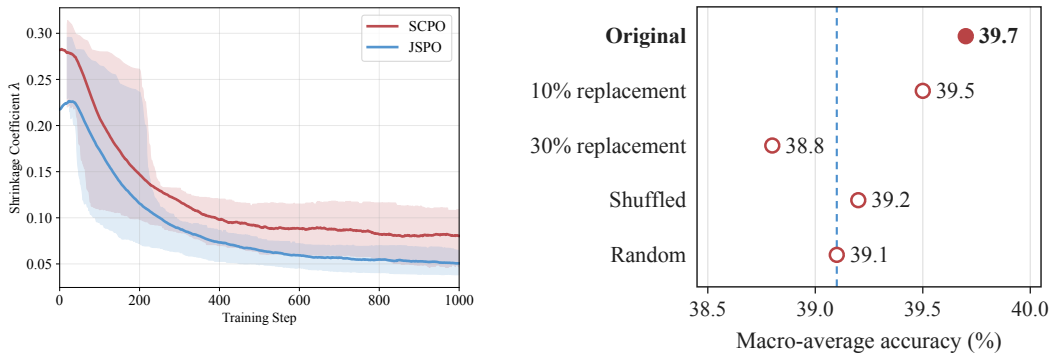
To further characterize how group-aware aggregation shapes the baseline, we report the mean shrinkage coefficient λ over training for both SCPO and JSPO. As shown in Figure 4a, SCPO maintains a consistently larger mean λ, placing more weight on its skill-group target than JSPO places on its batch-level target. Under Equation (7), lower within-group dispersion $\hat{S}_c^2$ increases the shrinkage weight for a fixed positive estimate of prompt-level sampling variance. Grouping prompts with similar values can therefore help SCPO rely more on a shared target when local reward estimates are noisy, while limiting the bias from shrinking toward a mismatched target. Appendix B.5 extends the weight comparison to 7B and compares grouping choices on matched 1.5B batches. Appendix B.4 uses the same batches to examine how the shrinkage target and weight affect baseline MSE.

4.4 Ablation Studies

The core design choice in SCPO is the prompt grouping that defines c, since it determines which prompts contribute to each shrinkage target. We examine how performance varies with the choice and granularity of grouping, and separately evaluate the history tracker, accuracy-bin sampling, and skill-label quality. We report single training runs throughout this section, with GRPO (38.4%, 44.8%) and JSPO (39.1%, 45.0%) as references at 1.5B and 7B.

Grouping strategy. A natural concern about SCPO is its sensitivity to the choice of grouping. We compare three ways to form groups (Table 2). **Subject groups** (coarse-grained, 7 groups) use the dataset-provided MATH subject categories (e.g., Algebra, Geometry, Number Theory) directly as group labels, requiring no clustering or LLM annotation. **k-means embedding groups** (e.g., $k = 64$ groups) cluster prompts via k-means over their text embeddings. **Skill groups** (default, $\sim$100 groups) use metacognitive skill labels [9] predicted by prompting an external LLM.

Robust to coarse grouping. With only seven MATH subject labels, SCPO (Subject) reaches 39.3% on 1.5B and 45.8% on 7B, outperforming the single-run GRPO references by 0.9 and 1.0 percentage points and the JSPO references by 0.2 and 0.8 points, respectively. The gap to fine-grained skill grouping is small at both scales (0.4 and 0.8 points). Even coarse subject categories can provide enough structure for within-group shrinkage to outperform the batch-level baseline.



(a) Mean shrinkage coefficient over training

(b) Accuracy under skill-label perturbations

Figure 4: Shrinkage dynamics and skill-label sensitivity for *Qwen2.5-Math-1.5B*. The shaded bands in (a) come from smoothing the coefficient traces. Each dot in (b) shows the macro-average from one training run, and the dashed line marks the single-run JSPO reference (39.1%).

Table 2: **Grouping, history tracking, and accuracy-bin sampling.** Accuracy (%) from one training run per configuration. All variants use $W = 75$ except the current-batch target ($W = 0$). Skill + Bins samples bins uniformly, and CL uses max-entropy bin selection. For binned variants, #Groups counts skill–bin pairs. The best and second-best distinct scores in each column and model block are **bolded** and underlined.

Configuration	#Groups	AIME 24	AIME 25	AMC 23	MATH-500	Minerva	Olympiad	Avg.
				Qwen2.5-Math-1.5B				
k-means	64	14.4	11.4	56.9	75.4	36.3	39.0	38.9
Subject	7	15.2	11.9	57.5	75.4	36.0	39.7	39.3
Skill ($W = 0$)	∼100	15.8	11.8	56.1	**77.2**	35.5	39.9	39.4
Skill ($W = 75$)	∼100	16.0	11.4	**58.3**	76.4	36.0	**40.4**	39.7
Skill + Bins	∼300	16.6	13.4	58.1	76.5	**36.4**	39.7	40.1
Skill + Bins + CL	∼300	**17.9**	**15.0**	57.8	76.5	**36.4**	39.8	**40.6**
				Qwen2.5-Math-7B				
k-means	64	31.2	15.0	60.6	81.1	42.6	43.7	45.7
Subject	7	28.0	**18.3**	61.8	80.0	42.0	44.5	45.8
Skill ($W = 0$)	∼100	32.0	14.9	62.9	80.8	42.1	44.2	46.2
Skill ($W = 75$)	∼100	**32.3**	15.8	63.5	81.2	41.5	**45.0**	46.6
Skill + Bins	∼300	32.0	16.2	64.5	81.8	43.6	44.6	47.1
Skill + Bins + CL	∼300	31.8	16.2	**67.3**	**83.5**	**44.1**	44.4	**47.9**

Embedding-based grouping underperforms. The k-means variant achieves 38.9% on 1.5B and 45.7% on 7B, below skill-based grouping at both scales (39.7% and 46.6%). One possible explanation is that embedding similarity can group topically related prompts with different solution procedures, leaving more heterogeneous groups and weakening the group-level baseline.

History-based group target. Holding skill grouping fixed, replacing the current-batch group mean ($W = 0$) with the sliding-window target ($W = 75$) raises the 1.5B average from 39.4% to 39.7% and the 7B average from 46.2% to 46.6%. The current-batch variant *already exceeds the single-run GRPO and JSPO references* at both scales. Pooling recent same-group rollouts can reduce noise in the group target while limiting the age of retained observations. The effect varies by benchmark, with slight declines on MATH-500 at 1.5B and Minerva at 7B.

Accuracy bins and curriculum. We split each skill group into accuracy terciles using the per-prompt accuracies from the tracker in Section 3.3. Relative to Skill, adding bins and sampling them uniformly improves the average by 0.4 and 0.5 percentage points at 1.5B and 7B, respectively. With the binning procedure and tracker settings fixed, max-entropy bin selection adds a further 0.5 and 0.8 points. This optional curriculum selects bin b with probability proportional to $w_b = \exp(-\alpha\,\mathrm{KL}(p_b\|0.5))$, where p_b is the bin's mean tracked accuracy, $\alpha = 2$, and the KL is between

Table 3: **Results on Science QA.** Accuracy (%) on ARC-Challenge, OpenBookQA, and held-out GPQA-Diamond. Each post-training method uses 400 steps with $M = 8$ rollouts per prompt. Avg. is the macro-average. The best and second-best post-training scores are **bolded** and underlined, respectively.

Method	Grouping	#Groups	ARC-C	OpenBookQA	GPQA-Diamond (OOD)	Avg.
			Qwen3-0.6B			
Base	–	–	69.3	64.4	23.3	52.3
GRPO [41]	–	–	75.1	78.1	25.0	59.4
JSPO [51]	Batch	1	75.3	78.3	26.8	60.1
SCPO (ours)	Skill	20	**76.5**	**79.8**	**28.5**	**61.6**

Bernoulli distributions. This rule favors bins near 50% accuracy and reuses the tracked accuracies without additional rollouts. Across both changes, AIME 24 and AIME 25 improve by 1.9 and 3.6 points at 1.5B, while AMC 23 and MATH-500 improve by 3.8 and 2.3 points at 7B.

Sensitivity to noisy skill labels. We retrain SCPO at 1.5B after perturbing the original skill labels (Figure 4b). For partial replacement, each prompt independently receives another randomly drawn prompt's full skill set with probability 0.1 or 0.3. Shuffling permutes the original skill sets across prompts, preserving group sizes and the distribution of label counts per prompt. The random control draws skills uniformly from the same vocabulary, with label counts drawn from the original distribution to keep the number of labels per prompt comparable.

Replacing labels at the 10% rate lowers the average by 0.2 points, while 30% replacement lowers it to 38.8%. Shuffled and random labels yield 39.2% and 39.1%, close to the single-run JSPO reference (39.1%). The shuffle contrast suggests that the prompt-to-skill assignment matters even when group sizes are held fixed. Under Equation (7), greater within-group dispersion reduces the shrinkage weight at a fixed prompt-level sampling variance, which can limit reliance on a less informative group target. Appendix B.3 gives the per-benchmark scores.

4.5 Scientific Multiple-Choice Reasoning

To evaluate SCPO in a second reasoning setting, we post-train Qwen3-0.6B on the combined training splits of ARC-Challenge [8] and OpenBookQA [36]. We follow the same annotation procedure as in the math experiments, using GPT-5-mini with a science-specific taxonomy of 20 skills. We evaluate on the in-distribution test splits and on GPQA-Diamond [39], which is held out from Science QA post-training. Full dataset construction, annotation pipeline, and training details are in Appendix B.2.

As shown in Table 3, SCPO improves average accuracy across the three benchmarks by 2.2 percentage points over GRPO and 1.5 over JSPO. *The largest gain is on GPQA-Diamond*, where the model post-trained with SCPO reaches 28.5% accuracy, 3.5 and 1.7 points above GRPO and JSPO, respectively. Together with the math experiments, these results show gains from applying SCPO in both reasoning settings, using the same shrinkage rule with domain-specific skill groups.

5 Conclusion

In this paper, we proposed **Skill-Coupled Policy Optimization (SCPO)**, a group-aware baseline estimation framework for critic-free RLVR that pools reward statistics across prompts with shared reasoning skills. SCPO applied *within-group shrinkage* to move prompt-level reward estimates toward group-consistent value targets, limiting the influence of unrelated prompts on the baseline. We instantiated SCPO with a James–Stein shrinkage baseline and introduced a lightweight history-based tracker to stabilize group targets under limited rollouts and nonstationary policies. Experiments on mathematical and scientific reasoning showed that SCPO improved macro-average accuracy over prompt-local and global shrinkage baselines across the model scales we evaluated. SCPO relies on informative prompt groups, and our evaluation covers only the Qwen model family (Appendix F). Looking ahead, richer group constructions (e.g., hierarchical skill structures) may further improve estimator fidelity. Moreover, the group value tracker provides a natural foundation for curriculum learning. Its prompt-level statistics support difficulty-based sampling within skill groups and could guide future sampling across groups as the policy's competence and uncertainty evolve.

Acknowledgements

This work is supported in part by the National Science Foundation (NSF) under grants CPS-2313110, ECCS-2143559, ECCS-2332060, ECCS-2318759 and ECCS-2531023.

References

[1] A. Ahmadian, C. Cremer, M. Gallé, M. Fadaee, J. Kreutzer, O. Pietquin, A. Üstün, and S. Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In L.-W. Ku, A. Martins, and V. Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand, Aug. 2024. Association for Computational Linguistics.

[2] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[3] S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

[4] Y. Chang, Y. Wu, and Y. Chang. BV-blend: Uncertainty-weighted historical baselines for stable critic-free RL with verifiable rewards. In M. Liakata, V. P. Moreira, J. Zhang, and D. Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 1706–1725, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[5] A. Chen, A. Li, B. Gong, B. Jiang, B. Fei, B. Yang, B. Shan, C. Yu, C. Wang, C. Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

[6] M. Chen, N. Roberts, K. Bhatia, J. Wang, C. Zhang, F. Sala, and C. Ré. Skill-it! a data-driven skills framework for understanding and training language models. *Advances in Neural Information Processing Systems*, 36:36000–36040, 2023.

[7] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. D. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[8] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[9] A. Didolkar, A. Goyal, N. R. Ke, S. Guo, M. Valko, T. Lillicrap, D. Jimenez Rezende, Y. Bengio, M. C. Mozer, and S. Arora. Metacognitive capabilities of LLMs: An exploration in mathematical problem solving. *Advances in Neural Information Processing Systems*, 37:19783–19812, 2024.

[10] X. Du, Y. Yao, K. Ma, B. Wang, T. Zheng, K. Zhu, M. Liu, Y. Liang, X. Jin, Z. Wei, et al. Supergpqa: Scaling LLM evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

[11] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[12] K. Han, Y. Zhou, M. Gao, G. Zhou, S. Li, A. Kumar, X. Fan, W. Li, and L. Zhang. Ebpo: Empirical bayes shrinkage for stabilizing group-relative policy optimization. *arXiv preprint arXiv:2602.05165*, 2026.

[13] Y. Hao, L. Dong, X. Wu, S. Huang, Z. Chi, and F. Wei. On-policy RL with optimal reward baseline. *arXiv preprint arXiv:2505.23585*, 2025.

[14] Y. Hao, L. Dong, X. Wu, S. Huang, Z. Chi, and F. Wei. On-policy RL with optimal reward baseline. *arXiv preprint arXiv:2505.23585*, 2025.

[15] C. He, R. Luo, Y. Bai, S. Hu, Z. Thai, J. Shen, J. Hu, X. Han, Y. Huang, Y. Zhang, et al. Olympiadbench: A challenging benchmark for promoting AGI with Olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, 2024.

[16] H. He, Y. Ye, Q. Cai, C. Hu, B. Jiao, D. Jiang, and L. Pan. Random policy valuation is enough for LLM reasoning with verifiable rewards. *arXiv preprint arXiv:2509.24981*, 2025.

[17] Y. He, A. Panigrahi, Y. Lin, and S. Arora. Adaptmi: Adaptive skill-based in-context math instruction for small language models. *arXiv preprint arXiv:2505.00147*, 2025.

[18] Y. He, A. Panigrahi, Y. Lin, and S. Arora. Skill-targeted adaptive training. *arXiv preprint arXiv:2510.10023*, 2025.

[19] Z. He, T. Liang, J. Xu, Q. Liu, X. Chen, Y. Wang, L. Song, D. Yu, Z. Liang, W. Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

[20] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[21] J. Hu, J. K. Liu, H. Xu, and W. Shen. Reinforce++: An efficient RLHF algorithm with robustness to both prompt and reward models. *arXiv preprint arXiv:2501.03262*, 2025.

[22] W. James, C. Stein, et al. Estimation with quadratic loss. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 361–379. University of California Press, 1961.

[23] L. Kong, X. Liu, G. Chen, M. Q. Ma, X. Song, Y. Sun, M. Yurochkin, T. W. Killian, R. Salakhutdinov, K. Zhang, E. P. Xing, and Z. Liu. From reasoning traces to reusable modules: Understanding compositional generalization in language model reasoning. In T. Zhang, M. Dudik, M. Jaggi, A. Agarwal, S. Li, D. Schuurmans, J. Zhu, F. Berkenkamp, H. Dong, and A. Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 60023–60050. PMLR, 06–11 Jul 2026.

[24] W. Kool, H. van Hoof, and M. Welling. Buy 4 REINFORCE samples, get a baseline for free!, 2019.

[25] H. Le, Y. Wang, A. D. Gotmare, S. Savarese, and S. C. H. Hoi. CodeRL: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21314–21328, 2022.

[26] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

[27] J. Li, L. Yu, Q. Cui, Z. Zhang, J. Zhou, Y. Ye, and C. Zhang. MASS: Mathematical data selection via skill graphs for pretraining large language models. In A. Singh, M. Fazel, D. Hsu, S. Lacoste-Julien, F. Berkenkamp, T. Maharaj, K. Wagstaff, and J. Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 36722–36742. PMLR, 13–19 Jul 2025.

[28] S. Li, Z. Zhou, W. Lam, C. Yang, and C. Lu. RePO: Replay-enhanced policy optimization. *arXiv preprint arXiv:2506.09340*, 2025.

[29] M. Liu, S. Diao, X. Lu, J. Hu, X. Dong, Y. Choi, J. Kautz, and Y. Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.

[30] Z. Liu, C. Chen, W. Li, P. Qi, T. Pang, C. Du, W. S. Lee, and M. Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[31] Z. Liu, J. Liu, Y. He, W. Wang, J. Liu, L. Pan, X. Hu, S. Xiong, J. Huang, J. Hu, et al. Part I: Tricks or traps? a deep dive into RL for LLM reasoning. *arXiv preprint arXiv:2508.08221*, 2025.

[32] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[33] X. Lv, Y. Zuo, Y. Sun, H. Liu, Y. Wei, Z. Chen, L. He, X. Zhu, K. Zhang, B. Wang, et al. Towards a unified view of large language model post-training. *arXiv preprint arXiv:2509.04419*, 2025.

[34] American invitational mathematics examination (AIME). Mathematics Competition Series, n.d.

[35] American mathematics competitions (AMC 10/12). Mathematics Competition Series, n.d.

[36] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium, Oct.-Nov. 2018. Association for Computational Linguistics.

[37] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[38] L. Phan, A. Gatti, Z. Han, N. Li, J. Hu, H. Zhang, C. B. C. Zhang, M. Shaaban, J. Ling, S. Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[39] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[40] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[41] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[42] A. Singh, A. Fry, A. Perelman, A. Tart, A. Ganesh, A. El-Kishky, A. McLaughlin, A. Low, A. Ostrow, A. Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

[43] C. Stein et al. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In *Proceedings of the Third Berkeley symposium on mathematical statistics and probability*, volume 1, pages 197–206, 1956.

[44] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[45] Z. Xu and Z. Ding. Single-stream policy optimization. *arXiv preprint arXiv:2509.13232*, 2025.

[46] J. Yan, Y. Li, Z. Hu, Z. Wang, G. Cui, X. Qu, Y. Cheng, and Y. Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025.

[47] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[48] A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[49] F. Yang, Z. Chen, X. Wang, X. Lu, J. Chai, G. Yin, W. Lin, S. Ma, F. Zhuang, D. Wang, et al. Your group-relative advantage is biased. *arXiv preprint arXiv:2601.08521*, 2026.

[50] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, W. Dai, T. Fan, G. Liu, L. Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[51] G. Zeng, Z. Zhou, D. Arora, and A. Zanette. Shrinking the variance: Shrinkage baselines for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2511.03710*, 2025.

[52] H. Zhang, J. Fu, J. Zhang, K. Fu, Q. Wang, F. Zhang, and G. Zhou. RLEP: Reinforcement learning with experience replay for LLM reasoning. *arXiv preprint arXiv:2507.07451*, 2025.

[53] J. Zhang, Z. Hao, W. Xia, H. Dong, H. Wang, C. Wei, Y. Zhou, Y. Qi, Q. Lin, and J. Cao. GAPO: Robust advantage estimation for real-world code LLMs. *arXiv preprint arXiv:2510.21830*, 2025.

[54] L. Zhang, Y. Zhang, W. Hu, and L. Wang. Skill-aware data selection and fine-tuning for data-efficient reasoning distillation. In M. Liakata, V. P. Moreira, J. Zhang, and D. Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 595–604, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[55] Y. Zhao, Y. Liu, J. Liu, J. Chen, X. Wu, Y. Hao, T. Lv, S. Huang, L. Cui, Q. Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

[56] X. Zhu, M. Xia, Z. Wei, W.-L. Chen, D. Chen, and Y. Meng. The surprising effectiveness of negative reinforcement in LLM reasoning. *arXiv preprint arXiv:2506.01347*, 2025.

Appendix

A Related Work **15**

B Additional Experiment Results. **16**

B.1 Heterogeneity of the RLVR Training Data 16

B.2 SCPO on Multiple-Choice Question Answering 16

B.3 Detailed Ablation Results. 17

B.4 Monte Carlo Analysis of Value-Estimation Error 18

B.5 Shrinkage Weights over Training 19

C Off-Policy Group Targets for James–Stein Baselines **19**

C.1 Setting and Notation 19

C.2 Group Shrinkage Baseline and MSE Objective 20

C.3 MSE Decomposition with Off-Policy Target 20

C.4 Estimating the Target-Error Terms 20

C.5 Weight Used in the Algorithm. 21

D Implementation Details **22**

D.1 Pseudocode for Computing Calibrated Group-wise Advantage in SCPO 22

E Experimental Details. **23**

E.1 Training Hyperparameters 23

E.2 Prompt Templates. 23

E.3 Skill Annotation on MATH dataset 24

F Limitations **26**

G Notation **28**

A Related Work

RL with verifiable rewards for reasoning post-training. Reinforcement learning has become a standard component in post-training LLMs, historically through PPO-style RLHF pipelines that learn both a reward model and a value-function critic [40, 37]. More recently, *reinforcement learning with verifiable rewards* (RLVR) has enabled critic-free scaling for reasoning tasks where correctness can be checked programmatically, motivating GRPO-style policy-gradient methods that estimate advantages from a small number of on-policy rollouts per prompt [41, 11]. A growing body of work studies algorithmic and systems refinements to make RLVR stable and efficient at scale, including large-scale open training recipes and infrastructure [50], off-policy guidance that mixes demonstrations with on-policy rollouts [46], and analyses of training pathologies and biases in GRPO-like objectives [30]. In parallel, many variants target stability, exploration, and robustness in long-CoT settings, such as geometric-mean objectives to reduce sensitivity to reward outliers [55], replay- or reuse-based improvements [28, 52], and robust advantage estimators in code and tool-use domains [53]. Our work is complementary to these efforts: we focus on the *statistical problem of baseline/value estimation* that underpins critic-free RLVR's variance and stability.

Critic-free baseline and value estimation in RLVR. Baseline design is a classical variance-reduction problem in policy gradients, where a control variate preserves unbiasedness while reducing gradient variance [44, 24]. In RLVR, this question becomes acute because each prompt receives

only a handful of sampled rollouts, making prompt-local baselines noisy. REINFORCE++ addresses this by using global batch statistics for normalization, arguing that large batch means and variances concentrate and stabilize training [21]. JSPO formalizes baseline construction as a joint mean-estimation problem and applies James–Stein shrinkage to interpolate between prompt-level estimates and a global batch target, with a leave-one-out construction to mitigate bias from baseline–reward coupling [51]. Other recent approaches seek improved or theoretically motivated baselines, including optimal reward baselines in on-policy RL [14] and single-stream policy optimization methods that restructure the estimation pipeline [45]. Yang et al. [49] analyze GRPO-style group-relative advantages and identify a systematic difficulty-dependent bias, proposing a history-aware reweighting scheme that mitigates this effect and boosts hard-benchmark performance. At a different extreme, Random Policy Valuation (ROVER) [16] argues that highly accurate on-policy value estimation may not be necessary for effective RLVR in certain reasoning regimes, shifting emphasis from precise policy evaluation toward maintaining diversity and exploration. Our contribution sits in the middle ground: we retain the policy-optimization framing but redesign the baseline to reflect *structured heterogeneity* in post-training data.

Concurrent work on empirical Bayes shrinkage. Empirical Bayes Policy Optimization (EBPO) [12] and SCPO share the intuition that pooling reward statistics can stabilize prompt-level value estimates with few rollouts, while pooling prompts of different difficulty can yield a mismatched shrinkage target. Both methods shrink prompt means toward a history-based target, with stronger shrinkage when sampling noise is large relative to dispersion across prompts. EBPO tracks a global prior with Welford's algorithm and explores topic- or difficulty-based ordering to better align that prior with the current prompts. SCPO maintains a separate sliding-window target for each skill group and estimates the dispersion from prompts sampled within that group.

B Additional Experiment Results

B.1 Heterogeneity of the RLVR Training Data

Figure 5 substantiates the value-heterogeneity claim from Section 1 with finer detail. To produce this analysis, we sample 8 responses per question from Qwen2.5-Math-7B-Instruct on the MATH dataset [20], compute the per-question average accuracy, and aggregate by the metacognitive skill labels introduced in Appendix E.3. The radial plot then reports the mean accuracy per skill cluster, with each axis labeled by its cluster name. Although the overall corpus-level accuracy is 0.76, skill-level accuracies span a 0.55-wide range: well-rehearsed procedural skills such as *log and exp* reach 0.96, whereas multi-step geometric reasoning skills like *triangle properties* remain at 0.41. This spread is far larger than what a single global baseline can capture, providing the empirical motivation for shrinking prompt-level estimates toward skill-coupled group targets rather than a uniform batch-level mean.

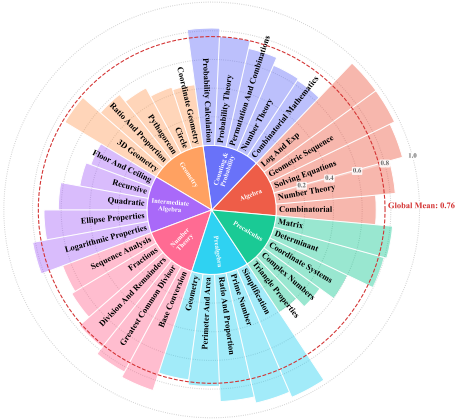


Figure 5: The skill accuracy distribution of Qwen2.5-Math-7B-Instruct on the MATH dataset [20]. While the average accuracy across the dataset is 0.76, skill-level performance varies substantially; top-performing skills such as *log and exp* reach 0.96, whereas lower-performing skills like *triangle properties* remain at 0.41.

B.2 SCPO on Multiple-Choice Question Answering

This appendix details the Science QA experiment in Section 4.5, including the dataset, science-specific skill annotations, and post-training setup.

Dataset. We construct a Science QA dataset by merging ARC-Challenge [8] and OpenBookQA [36], both standard 4-way MCQA benchmarks for elementary and middle-school science. The merged

Table 4: The 20 science-domain skill clusters used for SCPO grouping on the Science QA training set, sorted by frequency. Multi-label annotations: 74.3% of $6{,}076$ examples are tagged with more than one skill (average 1.85 skills per example).

Skill cluster	**Count**
animal_biology_ecology_and_organisms	1,575
ecology_environment_and_sustainability	1,148
chemistry_matter_and_materials	772
plant_biology_agriculture_and_environment	730
energy_heat_and_thermal_science	692
human_biology_health_and_medicine	666
weather_climate_and_atmospheric_science	585
human_behavior_movement_and_safety	572
forces_motion_mechanics_and_simple_machines	526
engineering_technology_and_tools	522
astronomy_space_and_earth_sun_systems	521
earth_science_geology_and_landforms	518
genetics_evolution_reproduction_and_life_cycles	467
measurement_geometry_and_quantitative_reasoning	415
scientific_inquiry_methods_and_tools	358
waves_light_sound_and_vision	356
oceans_water_and_hydrology	275
electricity_magnetism_and_circuits	227
atomic_molecular_and_cellular_biology	196
language_cognition_and_learning	99

corpus contains $6{,}076$ training examples ($1{,}119$ from ARC-Challenge, $4{,}957$ from OpenBookQA), 799 validation examples, and $1{,}672$ test examples ($1{,}172$ ARC-Challenge, 500 OpenBookQA).

Skill annotation. Following the same metacognitive-skill annotation pipeline used for MATH as shown in Appendix E.3, we prompt GPT-5-mini [42] to label each training example with up to five skills drawn from a pre-defined taxonomy of 20 science-domain clusters. The resulting annotations are multi-label: 74.3% of examples are tagged with more than one skill, with an average of 1.85 skills per example. The skill distribution is long-tailed: the most frequent cluster (*animal_biology_ecology_and_organisms*) contains $1{,}575$ examples, while the rarest (*language_cognition_and_learning*) contains only 99 (Table 4). This long-tail structure mirrors the MATH-side skill distribution and motivates the size-aware group sampling described in Section 3.1.

Training and evaluation. We post-train Qwen3-0.6B for 400 optimization steps using GRPO, JSPO, and SCPO with the same hyperparameter configuration as the math experiments ($M = 8$ rollouts per prompt, $G = 16$ groups per batch, $P = 8$ prompts per group; sliding-window history target with $W = 75$ for SCPO). Rewards are exact-match indicators on the predicted answer letter. We evaluate in two settings: (i) in-distribution test splits of ARC-Challenge ($1{,}172$ examples) and OpenBookQA (500 examples); (ii) GPQA-Diamond [39], a graduate-level science QA benchmark held out as an out-of-distribution (OOD) probe. The headline accuracies and per-benchmark breakdown are reported in Table 3 of Section 4.5.

B.3 Detailed Ablation Results

Table 5 gives the per-benchmark scores underlying the label-quality comparison in Figure 4b. Each configuration reports one training run.

Global-bin sampling control. We also use the tracker to form accuracy terciles over the full training set and apply max-entropy bin selection with JSPO's batch-level shrinkage baseline. At 1.5B, this configuration reaches 38.2%, compared with 39.1% for the JSPO reference. This comparison changes both the sampling distribution and the use of tracked accuracy statistics, so the score reflects the global-bin setup as a whole.

Table 5: **Skill-label controls.** Per-benchmark accuracy (%) for SCPO with different label assignments on *Qwen2.5-Math-1.5B*. JSPO is the single-run reference.

Labels	AIME 24	AIME 25	AMC 23	MATH-500	Minerva	Olympiad	**Avg.**
Original	16.0	11.4	58.3	76.4	36.0	40.4	39.7
10% replacement	17.5	12.2	58.8	74.6	35.7	38.4	39.5
30% replacement	14.7	11.6	55.2	77.4	34.9	39.1	38.8
Shuffled	14.8	15.2	53.8	77.4	34.6	39.3	39.2
Random	14.5	12.7	55.4	77.6	35.3	39.0	39.1
JSPO	14.9	12.9	56.9	75.6	35.6	38.9	39.1

B.4 Monte Carlo Analysis of Value-Estimation Error

Following the Monte Carlo evaluation of JSPO [51], we compare baseline estimators on the same batches from two Qwen2.5-Math-1.5B SCPO runs (33 checkpoints for seed 1 and 18 for seed 3). Each batch contains 16 skill groups of 8 prompts, with 8 logged rollouts per prompt. We estimate each prompt's current-policy value $\widehat{V}_i$ from 64 independent rollouts at the same checkpoint and subtract $\widehat{V}_i(1-\widehat{V}_i)/63$ from the squared error to correct for sampling noise in the reference. We average the resulting errors over rollouts and prompts, then weight checkpoints equally within each run.

We compare the SCPO training replay with JSPO recomputed on the same rollouts. The replay reconstructs the history-based group target and shrinkage weight from the logs.[2] The $W = 0$ diagnostic replaces the history-based target with the current-batch group mean under the same target convention. The batch-pooled diagnostics estimate the shrinkage weight from statistics pooled across the batch, first keeping the history-based target fixed and then using a leave-one-out group target. In the post hoc averaging diagnostics, we assign each prompt the arithmetic mean of the replayed SCPO weights within its batch or skill group, retaining each estimator's local mean and target.

Table 6: **Baseline MSE on fixed SCPO batches.** Values are percentage differences from the mean MSE of JSPO recomputed on the same batches, separately for each seed. Positive values indicate higher MSE.

Shrinkage target	**Weight rule**	**Seed 1**	**Seed 3**
SCPO history ($W = 75$)	Training replay	$+3.63$	$+4.17$
SCPO current batch ($W = 0$)	Recomputed per prompt	$+4.29$	$+3.64$
SCPO history	Batch-pooled statistics	-0.95	-1.93
SCPO history (leave-one-out)	Batch-pooled statistics	-0.87	-1.90
Post hoc averaging of replayed SCPO weights			
JSPO batch target	Batch mean	$+0.03$	$+0.36$
SCPO history	Batch mean	-1.39	-1.74
SCPO history	Skill-group mean	-0.71	-0.55

The replayed estimator has higher MSE than JSPO on both runs, and removing history moves the point estimates in opposite directions across seeds (Table 6). Pooling the weight statistics lowers MSE with the target fixed. Resampling checkpoints and groups leaves the direction relative to JSPO unresolved for the pooled leave-one-out comparison.

Adding target variance alone or together with a bias-corrected drift estimate (Appendix C) changed the excess by at most 0.19 percentage points on the same relative scale. An alternative drift penalty reduced the excess to 2.36% and 2.66% on the two runs.

At the same batch-mean weight, the baseline using SCPO's replayed group target has lower MSE point estimates than the baseline using JSPO's batch target on both seeds. With SCPO's target fixed, averaging weights within each skill group lowers MSE relative to the training replay while

[2]The replayed group target μ_c includes the focal prompt's current-batch rewards. The replay weight is $(P-1)/P$ times $\hat{v}/(\hat{v}+\hat{s})$, where $\hat{v}$ and $\hat{s}$ average the sampling variances $\hat{\sigma}_k^2$ and squared deviations $(\bar{r}(x_k)-\mu_c)^2$ over the other $P-1$ prompts. The weight is zero when both estimates vanish.

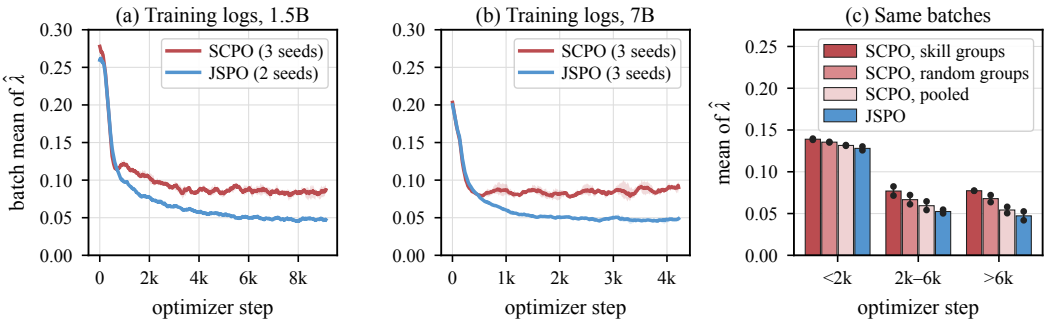


Figure 6: **Shrinkage weights in training logs and on matched batches.** (a, b) Mean and min–max range across seeds after centered smoothing over 33 RL iterations. (c) Mean weights on the same 1.5B SCPO batches, grouped by training stage and excluding the base checkpoint. Dots show the two seed means.

retaining differences in mean weight across groups. The averaged weights can depend on the focal prompt's rollouts through other prompts' weights, leaving open the roles of sampling noise and the leave-one-out rule. These post hoc diagnostics on fixed rollouts motivate *designing the skill-group target and shrinkage weight separately*, keeping the target local to a skill group while estimating the weight from a broader pool of prompts.

B.5 Shrinkage Weights over Training

We extend Figure 4a with training logs at both model scales and a comparison on the fixed 1.5B batches from Appendix B.4. In the training logs, the mean weight falls early for both methods and then remains higher for SCPO (Figure 6a, b). Over optimizer steps 6,001–8,000 at 1.5B and 3,001–4,224 at 7B, SCPO's mean weight is 1.74 and 1.83 times that of JSPO, respectively.

On the fixed batches, the mean weights on seeds 1 and 3 are 0.091 and 0.089 for the replayed skill groups, 0.080 and 0.082 for random regrouping, 0.071 and 0.073 for batch pooling, and 0.063 and 0.068 for JSPO (Figure 6c). Random regrouping reassigns the per-prompt variance and deviation terms to groups of eight while keeping each term centered on its original skill-group target. The lower weight under random regrouping therefore isolates the effect of group membership on the weight calculation at fixed group size. Appendix B.4 examines how batch pooling affects MSE.

C Off-Policy Group Targets for James–Stein Baselines

This section extends the James–Stein baseline analysis in JSPO [51] to a history-based group target. We derive the MSE-minimizing weight for a group under an idealized history-only target, then discuss how estimating the target-error terms affects the plug-in weight.

C.1 Setting and Notation

Consider a sampled group g containing P prompts and M rollouts per prompt. Let $r_{g,p,j}$ denote the reward of the j-th rollout for prompt $p \in \{1, \dots, P\}$, and define the prompt-level sample mean

$$\bar{r}_{g,p} := \frac{1}{M} \sum_{j=1}^{M} r_{g,p,j}. \tag{10}$$

Let $V_{g,p}$ denote the current-policy value (expected reward) for prompt (g, p)

$$V_{g,p} := \mathbb{E}_{\pi_t}[\, r \mid \text{prompt } (g,p)\,]\,. \tag{11}$$

We assume $\bar{r}_{g,p}$ is unbiased for $V_{g,p}$ with conditional sampling variance $\sigma^2_{g,p}$.

Each sampled group g has a group ID $c = c(g)$. Write $\mathbb{E}_c$, Var_c, and Cov_c for moments conditional on this group and the current policy. The current-policy group mean value and within-group heterogeneity are

$$\mu_c := \mathbb{E}_c[V_{g,p}], \qquad S_c^2 := \mathrm{Var}_c(V_{g,p}). \tag{12}$$

History-only target for the risk analysis. For the derivation, let $\tilde{\mu}_c$ be a group target computed from earlier batches. We model its moments as

$$\mathbb{E}_c[\tilde{\mu}_c] = \mu_c + b_c, \qquad \mathrm{Var}_c(\tilde{\mu}_c) = \tau_c^2, \tag{13}$$

where b_c is the drift bias from earlier policies and τ_c^2 is the sampling variance of the finite-history target. We assume that $\bar{r}_{g,p} - V_{g,p}$ has zero conditional mean given $V_{g,p}$ and $\tilde{\mu}_c$, and that $\mathrm{Cov}_c(\tilde{\mu}_c, V_{g,p}) = 0$.

C.2 Group Shrinkage Baseline and MSE Objective

We consider the within-group shrinkage form of Equation (5) with a common weight λ for the risk calculation

$$\hat{B}_{g,p}(\lambda) \;:=\; (1-\lambda)\,\bar{r}_{g,p} \;+\; \lambda\,\tilde{\mu}_c, \qquad \lambda \in [0,1]. \tag{14}$$

We average the value-estimation error over prompts in group c and their rollout randomness

$$\mathcal{R}_c(\lambda) \;:=\; \mathbb{E}_c\Big[\,\big(\hat{B}_{g,p}(\lambda) - V_{g,p}\big)^2\Big]. \tag{15}$$

Let $\bar{\sigma}_c^2 = \mathbb{E}_c[\sigma_{g,p}^2]$ be the average sampling variance of the prompt-level mean.

C.3 MSE Decomposition with Off-Policy Target

Lemma C.1 (Bias–variance decomposition for off-policy group targets)**.** *Under the assumptions above, the group-averaged MSE in Equation* (15) *is*

$$\mathcal{R}_c(\lambda) = (1-\lambda)^2\,\bar{\sigma}_c^2 + \lambda^2\big(S_c^2 + \tau_c^2 + b_c^2\big). \tag{16}$$

When the denominator is positive, the risk-minimizing common weight is

$$\lambda_c^\star = \frac{\bar{\sigma}_c^2}{\bar{\sigma}_c^2 + S_c^2 + \tau_c^2 + b_c^2}, \tag{17}$$

and satisfies $\mathcal{R}_c(\lambda_c^\star) \le \mathcal{R}_c(0) = \bar{\sigma}_c^2$. If the denominator is zero, every weight gives zero risk.

Proof. Expanding the error gives

$$\hat{B}_{g,p}(\lambda) - V_{g,p} = (1-\lambda)(\bar{r}_{g,p} - V_{g,p}) + \lambda(\tilde{\mu}_c - V_{g,p}). \tag{18}$$

The cross term vanishes because the prompt-level error is centered conditional on the prompt value and target. Averaging its squared error over the group gives $\bar{\sigma}_c^2$. For the target term, the zero-covariance assumption gives

$$\mathbb{E}_c[(\tilde{\mu}_c - V_{g,p})^2] = \mathrm{Var}_c(\tilde{\mu}_c) + \mathrm{Var}_c(V_{g,p}) + \big(\mathbb{E}_c[\tilde{\mu}_c] - \mathbb{E}_c[V_{g,p}]\big)^2 \tag{19}$$

$$= \tau_c^2 + S_c^2 + b_c^2, \tag{20}$$

which yields Equation (16). Minimizing this quadratic over $[0,1]$ gives Equation (17). □

Discussion. The group-averaged target error separates within-group heterogeneity S_c^2, finite-history variance τ_c^2, and squared policy drift b_c^2. At a fixed prompt, the target term is the expected squared error of the target conditional on that prompt's value. Equation (7) uses a prompt-specific sampling-variance estimate and shares the dispersion estimate across the group. The common-weight calculation isolates how uncertainty in the history-based target changes the risk.

C.4 Estimating the Target-Error Terms

The plug-in weight in Equation (7) estimates the sampling variance of each prompt mean as

$$\widehat{\sigma}_{g,p}^2 \;:=\; \frac{1}{M}\,\widehat{\mathrm{Var}}\big(\{r_{g,p,j}\}_{j=1}^M\big)\,. \tag{21}$$

It uses the variance of the observed prompt means as a proxy for within-group heterogeneity

$$\widehat{S}_c^2 \;:=\; \widehat{\mathrm{Var}}\big(\{\bar{r}_{g,p}\}_{p=1}^P\big)\,. \tag{22}$$

Drift bias proxy. A candidate drift proxy is the squared discrepancy between the current-batch group mean and the history-based target

$$\widehat{b}^2_{c,\mathrm{raw}} := \left(\bar{r}_g^{(\mathrm{cur})} - \tilde{\mu}_c\right)^2, \qquad \bar{r}_g^{(\mathrm{cur})} := \frac{1}{P}\sum_{p=1}^{P} \bar{r}_{g,p}. \tag{23}$$

The discrepancy includes sampling variation from both estimates. With an unbiased current-batch group mean, its expectation is

$$\mathbb{E}_c[\widehat{b}^2_{c,\mathrm{raw}}] = b_c^2 + \tau_c^2 + \mathrm{Var}_c(\bar{r}_g^{(\mathrm{cur})}) - 2\,\mathrm{Cov}_c(\bar{r}_g^{(\mathrm{cur})}, \tilde{\mu}_c).$$

With independent prompt draws and rollout noise, the current-batch variance is approximately $(S_c^2 + \bar{\sigma}_c^2)/P$. When the current and historical estimates are uncorrelated, adding the raw proxy alongside a separate estimate of τ_c^2 counts the target variance twice and also adds the current-batch sampling variance. A joint correction would first remove the sampling contribution from the squared discrepancy before truncating the drift estimate at zero. When the target shares rollouts with the current-batch estimate, the covariance term also enters this correction.

Target variance. With N_c equally weighted, approximately independent buffered prompt means of comparable variance, the variance of the target is approximately the per-entry variance divided by N_c. This gives a separate target-variance adjustment. Short or unevenly populated buffers and changing policies can make the estimate noisy.

Optional augmented weight. Including nonnegative estimates of both target-error terms gives

$$\widehat{\lambda}^{\mathrm{aug}}_{g,p} = \frac{\widehat{\sigma}^2_{g,p}}{\widehat{\sigma}^2_{g,p} + \widehat{S}^2_c + \widehat{\tau}^2_c + \widehat{b}^2_c}, \tag{24}$$

with zero weight if the denominator vanishes. Here $\widehat{b}^2_c$ denotes a drift estimate corrected for the sampling contribution when the two adjustments are combined. For fixed local sampling variance and dispersion, either nonnegative adjustment can only lower the weight. An inflated adjustment shifts the baseline toward the prompt-level mean even when the target remains informative.

C.5 Weight Used in the Algorithm

Algorithm 4 uses the two-term denominator in Equation (7), omitting both target-error adjustments. The sliding window in Section 3.3 limits the age of the observations used in the group target. A preliminary training comparison favored this rule over adding the drift proxy alone. Appendix B.4 separately measures baseline MSE on fixed rollouts, including a target-variance adjustment and a joint adjustment with a bias-corrected drift proxy.

D Implementation Details

D.1 Pseudocode for Computing Calibrated Group-wise Advantage in SCPO

The shrinkage weight in Algorithm 4 follows Equation (7). Its denominator combines the sampling variance of the prompt mean and the within-group dispersion. Appendix C discusses the additional target-variance and drift terms in the idealized risk.

Algorithm 1 SCPO Training Loop

Require: Groups $\{\mathcal{D}_g\}_{g\in\mathcal{G}}$, sampler $q(\cdot)$, policy π_θ
Require: Hyperparameters G, P, M, W, ϵ with $P, M > 1$, optional flag `LOO`
Initialize trackers $\{\mathcal{H}_g \leftarrow \emptyset,\ \mu_g \leftarrow 0\}_{g\in\mathcal{G}}$
for $t = 1, 2, \ldots$ **do**
  $(\{g_k\}, \{\mathcal{B}_{g_k}\}, \mathcal{B}) \leftarrow \textsc{BuildGroupedBatch}(\{\mathcal{D}_g\}, q, G, P)$
  $\{y^{(j)}(x), r^{(j)}(x)\} \leftarrow \textsc{CollectRollouts}(\pi_\theta, \mathcal{B}, M)$
  $\{\mu_g\} \leftarrow \textsc{UpdateTrackers}(\{g_k\}, \{\mathcal{B}_{g_k}\}, \{r^{(j)}(x)\}, \{\mathcal{H}_g\}, \{\mu_g\}, t, M, W)$
  $\{A^{(j)}(x)\} \leftarrow \textsc{ComputeSCPOAdv}(\{g_k\}, \{\mathcal{B}_{g_k}\}, \{r^{(j)}(x)\}, \{\mu_g\}, M, \texttt{LOO})$
  $\theta \leftarrow \text{GRPO}\textsc{Update}(\theta, \mathcal{B}, \{y^{(j)}(x)\}, \{A^{(j)}(x)\}, \epsilon)$
end for

Algorithm 2 BUILDGROUPEDBATCH

function $\textsc{BuildGroupedBatch}(\{\mathcal{D}_g\}, q, G, P)$
  Sample groups $\{g_k\}_{k=1}^{G} \sim q(\cdot)$
  $\mathcal{B} \leftarrow \emptyset$
  for $k = 1, \ldots, G$ **do**
    Sample prompts $\mathcal{B}_{g_k} \subset \mathcal{D}_{g_k}$ with $|\mathcal{B}_{g_k}| = P$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{B}_{g_k}$
  end for
  return $(\{g_k\}, \{\mathcal{B}_{g_k}\}, \mathcal{B})$
end function

Algorithm 3 UPDATETRACKERS (Sliding Window)

function $\textsc{UpdateTrackers}(\{g_k\}, \{\mathcal{B}_{g_k}\}, \{r^{(j)}(x)\}, \{\mathcal{H}_g\}, \{\mu_g\}, t, M, W)$
  for each sampled group $g \in \{g_k\}_{k=1}^{G}$ **do**
    $n_{g,t} \leftarrow |\mathcal{B}_g| \cdot M$
    $\hat{R}_{g,t} \leftarrow \frac{1}{n_{g,t}} \sum_{x\in\mathcal{B}_g} \sum_{j=1}^{M} r^{(j)}(x)$
    Append $(t, n_{g,t}, \hat{R}_{g,t})$ to $\mathcal{H}_g$; drop entries with $t' < t - W$
    $\mu_g \leftarrow \textsc{WindowMean}(\mathcal{H}_g)$
  end for
  return $\{\mu_g\}$
end function
function $\textsc{WindowMean}(\mathcal{H}_g)$
  return $\frac{\sum_{(t',n,\hat{R})\in\mathcal{H}_g} n\hat{R}}{\sum_{(t',n,\hat{R})\in\mathcal{H}_g} n}$
end function

Algorithm 4 COMPUTESCPOADV (Shrinkage Baseline + Advantages)

function COMPUTESCPOADV($\{g_k\}, \{\mathcal{B}_{g_k}\}, \{r^{(j)}(x)\}, \{\mu_g\}, M, \texttt{LOO}$)
  for each sampled group $g \in \{g_k\}_{k=1}^{G}$ **do**
    $\widehat{S}_g^2 \leftarrow$ WITHINGROUPDISPERSION($\mathcal{B}_g, \{r^{(j)}(x)\}, M$)
    for each prompt $x \in \mathcal{B}_g$ **do**
      $\bar{r}(x) \leftarrow \frac{1}{M}\sum_{j=1}^{M} r^{(j)}(x)$
      $\widehat{\sigma}^2(x) \leftarrow \frac{1}{M(M-1)}\sum_{j=1}^{M}\left(r^{(j)}(x) - \bar{r}(x)\right)^2$
      $d_x \leftarrow \widehat{\sigma}^2(x) + \widehat{S}_g^2$
      $\lambda_x \leftarrow \widehat{\sigma}^2(x)/d_x$ if $d_x > 0$, and 0 otherwise
      for $j = 1, \ldots, M$ **do**
        $b^{(j)}(x) \leftarrow$ SHRINKBASELINE($x, j, g, \lambda_x, \mu_g, \{r^{(j')}(x)\}, \texttt{LOO}$)
        $A^{(j)}(x) \leftarrow r^{(j)}(x) - b^{(j)}(x)$
      end for
    end for
  end for
  return $\{A^{(j)}(x)\}_{x \in \cup_g \mathcal{B}_g,\, j \in [M]}$
end function
function WITHINGROUPDISPERSION($\mathcal{B}_g, \{r^{(j)}(x)\}, M$)
  $\bar{r}(x) \leftarrow \frac{1}{M}\sum_{j=1}^{M} r^{(j)}(x)$ for all $x \in \mathcal{B}_g$
  $\hat{R}_g \leftarrow \frac{1}{|\mathcal{B}_g|}\sum_{x \in \mathcal{B}_g} \bar{r}(x)$
  return $\frac{1}{|\mathcal{B}_g|-1}\sum_{x \in \mathcal{B}_g}\left(\bar{r}(x) - \hat{R}_g\right)^2$
end function
function SHRINKBASELINE($x, j, g, \lambda_x, \mu_g, \{r^{(j')}(x)\}, \texttt{LOO}$)
  if LOO **then**
    $\bar{r}^{-j}(x) \leftarrow \frac{1}{M-1}\sum_{j' \neq j} r^{(j')}(x)$
    return $(1-\lambda_x)\bar{r}^{-j}(x) + \lambda_x \mu_g$
  else
    $\bar{r}(x) \leftarrow \frac{1}{M}\sum_{j'=1}^{M} r^{(j')}(x)$
    return $(1-\lambda_x)\bar{r}(x) + \lambda_x \mu_g$
  end if
end function

E Experimental Details

E.1 Training Hyperparameters

Following Dr.GRPO [30], we use Levels 3–5 questions from MATH [20] as the training dataset, which contains 8,523 mathematical problems. We utilize the AdamW [32] optimizer with a learning rate of 10^{-6}. For rollouts, the prompt batch size is $B = 128$ and we sample $M = 8$ responses per prompt by default. During each RL training round, the old policy $\pi_{\theta_{\text{old}}}$ produces 1,024 rollouts, and the current policy π_θ is updated 8 times with a batch size of 128. To construct a batch of $B = 128$ prompts at every iteration, as explained in Sec. 3.1, we sample $G = 16$ groups proportionally to their sizes and then randomly sample $P = 8$ prompts within each group. The sliding-window history target uses a window of $W = 75$ RL steps, which roughly corresponds to one epoch in our setting. Due to computing resource limitations, we train 1,000 steps for Qwen2.5-Math-1.5B and 500 steps for Qwen2.5-Math-7B to ensure convergence.

E.2 Prompt Templates

Following the best practice from previous works [30, 56], we adopt the Qwen-Math template for fine-tuning, as shown in Table E.2

Qwen-Math Template

```
<|im_start|>system
Please reason step by step, and put your final answer within
\\boxed{}.<|im_end|>
<|im_start|>user
{question}<|im_end|>
<|im_start|>assistant
```

E.3 Skill Annotation on MATH dataset

As discussed in Section 4.1, we adopt the skill taxonomy of Didolkar et al. [9] and use GPT-5-mini to label skills for the MATH dataset. Table 7 and Table 8 enumerate the full set of skills used for annotation, which we take directly from Didolkar et al. [9]. Following He et al. [17], we provide the LLM with this predefined skill set and prompt it to read each question and return up to five skills required to solve it. We provide an example prompt for annotating algebra questions below.

Example skill annotation prompt for MATH Algebra questions

[TASK]
You'll be given a math question. Your task is to output:
(1) < skill> list here up to five skill(s) that are required to solve this problem, seperated by commas </skill>.
(2) <reason> reason here why these skills are needed </reason>.

[SKILL LIST]
You should only choose the skills from this list:
[
"algebraic_expression_skills",
"algebraic_manipulation_skills",
"arithmetic_skills",
"calculation_and_conversion_skills",
"combinatorial_operations_and_basic_arithmetic",
"complex_number_skills",
"distance_and_midpoint_skills",
"exponent_and_root_skills",
"factoring_skills",
"function_composition_skills",
"function_skills",
"geometric_sequence_skills",
"graph_and_geometry_skills",
"inequality_skills",
"logarithmic_and_exponential_skills",
"number_theory_skills",
"polynomial_skills",
"quadratic_equation_skills",
"ratio_and_proportion_skills",
"sequence_and_series_skills",
"solving_equations"
]

[QUESTION]
{question}

[REASON AND SKILL(S)]

Subject	List of Skills
Per subject split in MATH	
Algebra	algebraic_expression_skills, algebraic_manipulation_skills, arithmetic_skills, calculation_and_conversion_skills, combinatorial_operations_and_basic_arithmetic, complex_number_skills, distance_and_midpoint_skills, exponent_and_root_skills, factoring_skills, function_composition_skills, function_skills, geometric_sequence_skills, graph_and_geometry_skills, inequality_skills, logarithmic_and_exponential_skills, number_theory_skills, polynomial_skills, quadratic_equation_skills, ratio_and_proportion_skills, sequence_and_series_skills, solving_equations
Counting and Probability	calculating_and_understanding_combinations, combinatorial_mathematics, combinatorics_knowledge, counting_principals, factorials_and_prime_factorization, number_theory_and_arithmetic_operations, permutation_and_combinations, probability_calculation_with_replacement, probability_concepts_and_calculations, probability_theory_and_distribution, understanding_and_applying_combinatorics_concepts
Geometry	3d_geometry_and_volume_calculation_skills, algebraic_skills, area_calculation_skills, circle_geometry_skills, combinatorics_and_probability_skills, coordinate_geometry_and_transformation_skills, other_geometric_skills, pythagorean_skills, quadrilateral_and_polygon_skills, ratio_and_proportion_skills, triangle_geometry_skills, trigonometry_skills, understanding_circle_properties_and_algebraic_manipulation

Table 7: List of skills used for annotating questions in each subject of the MATH dataset

Subject	List of Skills
	Per subject split in MATH
Intermediate Algebra	absolute_value_skills, algebraic_manipulation_and_equations, calculus_optimization_skills, complex_number_manipulation_and_operations, function_composition_and_transformation, graph_understanding_and_interpretation, inequality_solving_and_understanding, polynomial_skills, properties_and_application_of_exponents, quadratic_equations_and_solutions, recursive_functions_and_sequences, sequence_and_series_analysis_skills, simplification_and_basic_operations, solving_inequalities, solving_system_of_equations, summation_and_analysis_of_series, understanding_and_application_of_functions, understanding_and_applying_floor_and_ceiling_functions, understanding_and_manipulation_of_rational_functions, understanding_and_utilizing_infininte_series, understanding_ellipse_properties, understanding_logarithmic_properties_and_solving_equations
Number Theory	arithmetic_sequences, base_conversion, basic_arithmetic, division_and_remainders, exponentiation, factorization, greatest_common_divisor_calculations, modular_arithmetic, number_manipulation, number_theory, polynomial_operations, prime_number_theory, sequence_analysis, solving_equations, understanding_of_fractions
Pre-algebra	average_calculations, basic_arithmetic_operations, circles, counting_and_number_theory, exponentiation_rules, fractions_and_decimals, geometry, multiples_and_zero_properties, multiplication_and_division, perimeter_and_area, prime_number_theory, probability_and_combinatorics, ratio_and_proportion, solving_linear_equation
Pre-calculus	algebra_and_equations, basic_trigonometry, calculus, complex_number_operations, complex_numbers, coordinate_systems, determinant_calculation, geometric_relations, geometry_and_space_calculation, geometry_triangle_properties, matrix_operations, parametric_equations, sequences_series_and_summation, three_dimensional_geometry, trigonometric_calculations, vector_operations

Table 8: List of skills used for annotating questions in each subject of the MATH dataset (continued from Table 7)

F Limitations

While SCPO consistently improves over GRPO and JSPO across the settings we evaluate, several limitations are worth noting.

Reliance on external skill annotation. Our default grouping pipeline labels each prompt with up to five metacognitive skills using a strong proprietary LLM (GPT-5-mini). The resulting groups are only as informative as the annotator: noisier or less expressive labels would weaken the within-group shrinkage signal. We partially mitigate this in Section 4.4 by showing that SCPO is robust across grouping granularities — coarse 7-subject groups (free dataset metadata, no LLM annotation needed) still outperform GRPO and match JSPO — but tasks lacking any pre-defined skill taxonomy or accessible LLM annotator may require additional engineering to construct usable groups.

Empirical scope. We validate SCPO on mathematical reasoning (Qwen2.5-Math-1.5B/7B on the MATH-derived benchmark suite) and multiple-choice scientific reasoning (Qwen3-0.6B on Science QA + GPQA-Diamond). All experiments use the Qwen model family; code generation, agentic tool use, and other reasoning settings remain to be tested. We expect the framework to transfer in principle, since within-group shrinkage only requires (i) a meaningful prompt grouping and (ii) reward-labeled rollouts, but empirical confirmation in those domains is left to future work.

Statistical reporting. Table 1 reports standard deviations across three evaluation seeds for fixed checkpoints and three training seeds for post-trained models. The ablations and Science QA results use one training run per configuration.

Memory overhead of the history tracker. The sliding-window group value tracker stores recent prompt-rollout tuples per group within a window of W training steps. Memory scales linearly with the number of groups and the window size; for the configurations we use ($\sim$100 skill groups, $W = 75$), the overhead is negligible relative to model and rollout buffers, but very fine-grained groupings (e.g., $\sim$300 sub-groups under skill + accuracy-bins) increase the bookkeeping cost roughly proportionally.

G Notation

Table 9: Summary of notation used in the paper.

Symbol	Description
	Prompts, rollouts, and policy updates
$\mathcal{D}$	Training distribution over prompts.
x_i, y_i^j, r_i^j	The i-th prompt, its j-th rollout, and that rollout's verifier reward.
$\pi_\theta, \pi_{\theta_{\text{old}}}$	Current policy and policy used to collect rollouts, respectively.
$J(\theta), \mathcal{J}_{\text{GRPO}}(\theta)$	Expected-reward objective and clipped GRPO surrogate, respectively.
$s_{i,j}^t(\theta)$	Current-to-rollout policy probability ratio at token position t in y_i^j.
ϵ, ε	Policy-ratio clipping parameter and normalization guard, respectively.
	Prompt grouping and batch sampling
n, m	Number of prompts per batch and rollouts per prompt in Section 2.
$C, \mathcal{C}$	Number of groups in the training corpus and the collection of their group IDs.
$c, c(x)$	A group ID and the set of group IDs associated with prompt x, respectively.
G, P, M, B	Sampled groups per iteration, prompt draws per group, rollouts per prompt, and total prompt draws $B = GP$.
$\mathcal{B}_t, \mathcal{B}_c$	The batch at iteration t and the P prompts drawn from group c.
	Baseline estimates and shrinkage weights
μ_i, σ_i	Sample mean and standard deviation of rollout rewards for prompt x_i in GRPO.
$\hat{\mu}_i, \hat{\mu}_i^{-j}$	Full and leave-one-out prompt mean rewards in JSPO.
$\hat{\bar{\mu}}, \hat{\bar{\mu}}_{-i}$	JSPO's full batch mean and batch target excluding prompt x_i.
$\hat{v}_{-i}, \hat{s}_{-i}$	JSPO's estimates of prompt-mean sampling noise and cross-prompt dispersion, excluding x_i.
$\bar{r}(x_i), \bar{r}^{-j}(x_i)$	SCPO's full prompt mean reward and mean excluding rollout j.
$\hat{\sigma}_i^2, \hat{S}_c^2$	Estimated sampling variance of $\bar{r}(x_i)$ and sample variance of the P prompt means in $\mathcal{B}_c$.
$\bar{r}_c^{\text{cur}}$	Mean of the P prompt means in $\mathcal{B}_c$, used to center $\hat{S}_c^2$.
$\hat{\lambda}_i, \hat{\lambda}_i^{\text{c}}$	Estimated weights on the JSPO batch target and the SCPO group target, respectively.
$b_{i,j}^{\text{JS}}, b_{i,j}^{\text{SCPO}}$	Leave-one-out baselines for rollout j of prompt x_i under JSPO and SCPO.
$A_{i,j}$	SCPO advantage $r_i^j - b_{i,j}^{\text{SCPO}}$.
	History-based group targets and curriculum
$\hat{R}_{c,t}, n_{c,t}$	Mean reward and count of current-batch rollouts from prompts with membership c at iteration t.
$W_t, \mathcal{W}(t)$	Window length and the set of iterations it covers. W denotes the configured horizon.
$\mu_c(t), \mu_{c,-i}$	Tracked group target (also μ_c in Section 3) and its variant excluding x_i's current-batch rollouts.
p_b, w_b, α	Mean tracked accuracy of bin b, its sampling weight, and the weighting scale.
	Notation for the appendix analyses
$g, p, c(g)$	Sampled group, prompt index within it, and group ID in Appendix C.
$V_{g,p}, \widehat{V}_i$	Current-policy expected reward for prompt (g, p) and the 64-rollout reference estimate for x_i in Appendix B.4.
$\mu_c, \tilde{\mu}_c$	In Appendix C, the current-policy group mean and the history-only target.
$\sigma_{g,p}^2, \bar{\sigma}_c^2$	Sampling variance of the prompt mean and its average over prompts in group c.
S_c^2, τ_c^2, b_c	Variance of prompt values within the group, sampling variance of the history-only target, and signed drift bias.
$\mathcal{R}_c(\lambda), \lambda_c^\star$	Group-averaged baseline MSE and its minimizing common weight (Lemma C.1).
$\widehat{\lambda}_{g,p}^{\text{aug}}$	Optional weight with estimates of target variance and squared drift bias in Equation (24).
$\hat{v}, \hat{s}$	In Appendix B.4, the mean sampling variance and mean squared target deviation over the other $P - 1$ prompts.
$\mathcal{H}_g, q(\cdot)$	History buffer for group ID g and distribution for sampling groups in Algorithm 1.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 describe SCPO as a group-aware baseline estimation framework for critic-free RLVR. The three contributions listed in the introduction — empirically demonstrating skill-conditioned value heterogeneity, proposing the within-group shrinkage framework with a James–Stein instantiation, and introducing the history-based group value tracker — are addressed in Section 3 and empirically validated in Section 4.1 and Appendix B.2.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated Limitations section (Appendix F) discusses the reliance on an external LLM annotator for skill labels, the empirical scope (Qwen-family models on math and MCQA reasoning), the scope of multi-seed reporting, and the linear memory overhead of the sliding-window history tracker.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The James–Stein instantiation in Section 3 cites the foundational results of Stein and James–Stein together with the recent JSPO derivation [43, 22, 51]. Appendix C extends the derivation to the SCPO setting with explicit assumptions (approximate independence between the sliding-window historical target and the current-batch estimator), states the MSE objective, and derives the closed-form optimal shrinkage coefficient.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Datasets (Levels 3–5 of MATH for training; AIME 24/25, AMC 23, MATH-500, Minerva, OlympiadBench for evaluation; ARC-Challenge + OpenBookQA for MCQA), base models (Qwen2.5-Math-1.5B/7B, Qwen3-0.6B), batch construction ($G = 16$, $P = 8$, $M = 8$), and the evaluation protocol are reported in Section 4.1; the optimizer settings, training step counts, and sliding-window configuration are detailed in Appendix E.1; the GPT-based skill-annotation prompt template is provided in Appendix E.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We release an anonymized training and evaluation codebase, together with the GPT-annotated skill labels for both MATH (training and evaluation splits) and Science QA (ARC-Challenge + OpenBookQA), as supplemental material. The supplement includes a README with the exact commands to reproduce the main results, the recipe YAML configurations for each method, and pointers to the underlying public datasets.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 reports the headline experimental setup (models, benchmarks, evaluation protocol, batch construction). Appendix E.1 provides the complete training hyperparameters (AdamW with learning rate 10^{-6}, $G = 16$ groups per batch, $P = 8$ prompts per group, $M = 8$ rollouts per prompt, sliding-window size $W = 75$, 1,000 training steps for the 1.5B model and 500 for the 7B model). Cross-domain settings for Science QA are specified in Appendix B.2.

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Table 1 reports means and standard deviations over three seeds for each benchmark and for the per-seed macro-average. The standard deviations capture evaluation variability for fixed checkpoints and training variability for post-trained models. The ablation and Science QA results use one training run per configuration (Appendix F).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments are run on a single node equipped with 4 NVIDIA H100 GPUs. The full set of training runs reported in the paper — math experiments on Qwen2.5-Math-1.5B/7B (main results, grouping-strategy ablations, history-target ablations) and the cross-domain MCQA experiment on Qwen3-0.6B — consumes approximately two weeks of total GPU-hours on this node.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available datasets and open-weight pretrained models, involves no human subjects or private data, and conforms to the NeurIPS Code of Ethics. We have reviewed the Code of Ethics and identified no deviations.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper presents a methodological refinement to baseline estimation in critic-free RLVR for LLM post-training, instantiated as a drop-in replacement for the per-prompt mean baseline. The work is methodological, and the released artifacts are training code and skill annotations rather than a new model release; we do not foresee specific positive or negative societal impacts beyond those already associated with general LLM reasoning research.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

Justification: The released artifacts are training and evaluation code and GPT-annotated skill labels for the publicly available MATH and Science QA datasets. We do not release a new pretrained model or scraped dataset that would carry a high risk of misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets used (MATH [20], AIME 24/25, AMC 23, MATH-500, Minerva [26], OlympiadBench [15], ARC-Challenge [8], OpenBookQA [36], GPQA-Diamond [39]) and models (Qwen2.5-Math-1.5B/7B and Qwen3-0.6B [48, 47]) are publicly available and used under their respective open licenses. Original sources are cited in Section 4.1 and Appendix B.2.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We release training and evaluation code, the GPT-5-mini–generated metacognitive-skill annotations for the MATH dataset, and the corresponding skill annotations for the Science QA training set. The supplemental zip includes a README documenting the dataset schema, annotation prompt templates (also reproduced in Appendix E.3), recipe YAML configurations, and reproduction commands.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.

- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects research, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We use GPT-5-mini [42] to label each MATH and Science QA prompt with up to five metacognitive skills, which serve as the group labels for SCPO's within-group shrinkage (Section 4.1). The annotation pipeline follows the prompt template provided in Appendix E.3. The LLM is used only for offline label generation; it is not part of the trained policy or any inference-time component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.