RUC-SDPO: REWARD-UPLIFT-CALIBRATED SELF-DISTILLATION POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) enables scalable post-training, with Group Relative Policy Optimization (GRPO) widely used to translate verifier outcomes into policy updates. However, GRPO assigns the same group-relative advantage to every token in a trajectory, resulting in coarse credit assignment; its reward-driven update vanishes when all trajectories in a group receive identical rewards. Privileged on-policy self-distillation can provide dense token-level guidance from training-only context, but applying it indiscriminately may impose redundant, noisy, or reward-irrelevant preferences. We introduce Reward-Uplift-Calibrated Self-Distillation Policy Optimization (RUC-SDPO), a reward-anchored extension of GRPO that calibrates privileged self-distillation by the estimated task-level utility of the privileged context. Specifically, RUC-SDPO retains GRPO as the primary reward anchor on every sample, and uses reward-uplift calibration to estimate the verifier-reward uplift of privileged conditioning over deployment-matched conditioning. It maps only positive uplift to a bounded detached weight, and scales token-level distillation accordingly. The method requires no separate teacher model, learned reward model, or step-level annotations. Across mathematical reasoning and tool-call generation with Qwen3-4B, Qwen3-8B, and Qwen3-14B, RUC-SDPO outperforms a strengthened GRPO baseline, attaining the best average accuracy in all nine mathematical reasoning model–benchmark settings and the best score in eight of the nine tool-call settings.

1 INTRODUCTION

Large language models are increasingly post-trained via reinforcement learning with verifiable rewards (RLVR). In this paradigm, task verifiers programmatically assess output correctness to provide feedback. Among methods that leverage such feedback, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) has become a widely adopted on-policy approach: it samples multiple trajectories from the current policy, compares their verifier rewards within the group, and uses the resulting group-relative advantages to weight policy-gradient updates, thereby anchoring optimization in verifier-derived trajectory-level rewards.

However, like other outcome-level reinforcement learning methods (Uesato et al., 2023), GRPO provides a coarse learning signal. Within each trajectory, the same trajectory-level scalar advantage multiplies the policy-gradient contribution of every generated token, and it thus does not identify which token-level decisions are responsible for the outcome. The signal can also vanish entirely when all sampled trajectories within a group receive identical rewards, in which case all group-relative advantages become zero and produce no reward-driven policy gradient. This often occurs with sparse or binary verifiers and on problems that are uniformly easy or hard (Yu et al., 2025). These two failure modes motivate a denser learning signal that persists even when group rewards are identical.

To complement GRPO's coarse or vanishing trajectory-level signal with dense token-level guidance, recent work has explored on-policy self-distillation (Hübotter et al., 2026; Zhao et al., 2026), which exploits an asymmetry between training and deployment: some information is available during training but not at deployment, such as sibling rollouts for the same problem and their verifier outcomes. These methods package this training-only information into a privileged context that defines a privileged teacher view of the same policy, whose per-token distribution provides token-level targets for the

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

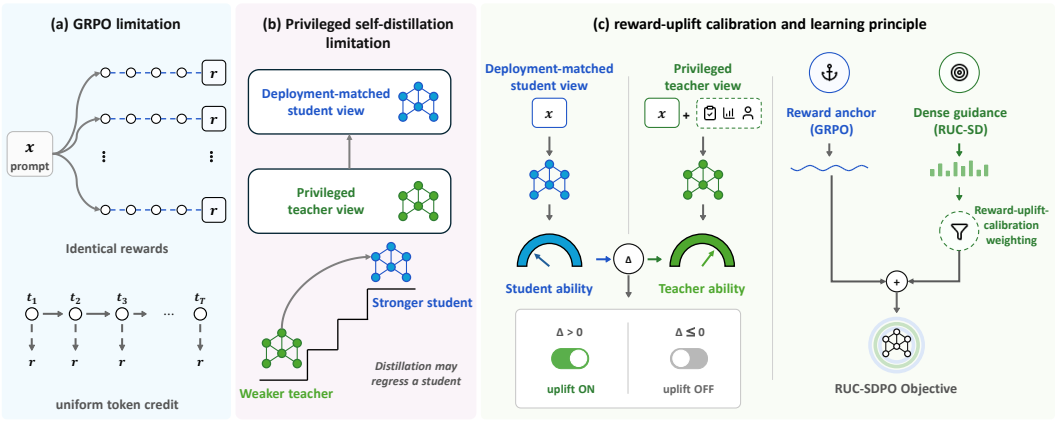


Figure 1: Motivation and key idea of RUC-SDPO. (a) GRPO assigns the same trajectory-level advantage to every token, and its signal vanishes when group rewards are identical. (b) Privileged self-distillation can regress a student that already surpasses its privileged teacher. (c) RUC-SDPO estimates the reward uplift of the privileged context and adds dense token-level guidance to the GRPO reward anchor only when the uplift is positive.

deployment-matched student view on the policy's own rollouts. This supplies fine-grained guidance beyond scalar trajectory rewards and can provide a learning signal even in identical-reward groups.

Yet naively distilling from the privileged teacher is not always beneficial. The guidance provided by the privileged teacher can be redundant when the student has already learned the target behavior or unreliable when the privileged context is noisy (Li et al., 2026). Directly matching the teacher distribution can also impose reward-irrelevant token-level preferences within trajectories that already satisfy the verifier. The central design problem is therefore when and how strongly privileged distillation should be applied. Existing approaches regulate this decision with indirect criteria, such as observed correctness, inter-sample agreement, teacher confidence, or signal specificity (Li et al., 2026; Stein et al., 2026; Zheng et al., 2026; Lu et al., 2026; Shen et al., 2026), none of which directly measures the incremental task-level utility contributed by privileged conditioning relative to deployment-matched conditioning.

In this paper, we propose Reward-Uplift-Calibrated Self-Distillation Policy Optimization (RUC-SDPO), which augments GRPO with a reward-uplift-calibrated distillation loss. Our method calibrates privileged distillation by task-level utility, measured as the expected verifier-reward difference between privileged and deployment-matched conditioning. This utility-based view leads to a reward-anchored design in which privileged distillation complements reward optimization only when privileged conditioning is estimated to improve verifier reward. Figure 1 summarizes the coarse or vanishing GRPO signal, the risk that privileged distillation regresses a student that already surpasses its privileged teacher, and our reward-anchored design that addresses both. In summary, our contributions are as follows:

- We propose RUC-SDPO, which retains GRPO as the primary reward anchor on every sample and treats privileged on-policy self-distillation as an auxiliary source of dense token-level guidance where the group-relative signal is coarse or vanishing.
- We design reward-uplift calibration as a bounded detached sample-level weight that activates distillation only for positive estimated uplift, so that whether distillation is applied is determined by verifier reward.
- We evaluate RUC-SDPO across three model scales and two task families, covering nine model–benchmark settings per family. RUC-SDPO attains the best average accuracy in all nine mathematical reasoning settings and the best score in eight of the nine tool-call settings.

2 Related Work

Reinforcement learning with verifiable rewards. Post-training large language models with RLVR has become a common approach for improving reasoning and tool use (Wei et al., 2022; Wang et al., 2023; Cobbe et al., 2021; Uesato et al., 2023). GRPO (Shao et al., 2024) replaces a learned value function with group-relative advantages computed from multiple sampled trajectories, making on-policy optimization practical at scale. Subsequent refinements stabilize this objective: DAPO (Yu et al., 2025) introduces asymmetric clipping and zero-variance-group filtering, analyses of R1-Zero-style training show that removing reward standard-deviation normalization mitigates length and difficulty biases (Liu et al., 2025b), and scaling studies add token-level importance-sampling correction for rollout-policy mismatch (Khatri et al., 2025). These outcome-level refinements leave reward granularity unchanged, the advantage thus stays uniform across the tokens of a trajectory and vanishes altogether when group rewards are identical. RUC-SDPO in contrast retains GRPO as a reward anchor but adds denser, token-level guidance where the group-relative signal becomes coarse or vanishing.

On-policy self-distillation for language models. A complementary line of work seeks dense learning signals by distilling a policy into itself along its own rollouts (Zelikman et al., 2022). Building on knowledge distillation (Hinton et al., 2015), recent methods cast policy improvement as on-policy self-distillation, where a privileged or hindsight-informed view supplies token-level targets for a deployment-matched view (Hübotter et al., 2026; Zhao et al., 2026). The teacher and student views share the same policy and differ only in training-time context, which avoids a separately trained teacher model. The paradigm has been extended to multi-turn agentic settings, where sibling rollouts and environment feedback form the privileged context (Lu et al., 2026), and combined with group-relative optimization by routing each sample to one of the two objectives (Li et al., 2026). The latter is the closest setting to ours, but it routes each sample exclusively to either group-relative optimization or distillation according to observed correctness, and a sample assigned to distillation receives no reward-driven update. Our RUC-SDPO instead keeps GRPO on every sample and treats calibrated distillation as an auxiliary term, which preserves the reward anchor everywhere and varies only the strength of dense guidance.

Regulating the distillation signal. Since privileged teachers can provide redundant, noisy, or misleading guidance, existing work further explores strategies for adjusting how strongly the privileged signal is trusted. Representative mechanisms include consensus gating (Stein et al., 2026), correctness-routed perplexity weighting (Zheng et al., 2026), token-level modulation for self-distilled agent learning (Lu et al., 2026), input-specific debiasing that corrects generic correlations in the distilled signal (Shen et al., 2026), and outcome-based gates that apply distillation only to all-failed prompts (Ding, 2026) or positive-advantage samples (Liu et al., 2026). These mechanisms establish that distillation should be selectively regulated rather than applied uniformly. Their criteria, however, are each computed from a single conditioning, which is either the observed outcome of a rollout or an intrinsic property of the teacher's own output distribution. Such a quantity cannot measure the verifier-reward uplift of the privileged context, and thus grants full trust even on samples where that context makes no difference under the verifier. Our RUC-SDPO instead calibrates trust by the task-level reward uplift induced by the privileged context, the strength of distillation thus tracks the estimated reward uplift.

3 Method

Figure 2 illustrates the overall framework of RUC-SDPO. It combines a GRPO reward anchor with reward-uplift-calibrated privileged self-distillation. Given an input x, we first sample a group of G on-policy rollouts and evaluate them with the verifier. The same rollout group serves two roles: its rewards define the GRPO reward anchor, while non-self sibling rollouts and their verifier outcomes form a leave-one-out privileged context $c^{(i)}$ for each target rollout $y^{(i)}$. RUC-SDPO then evaluates a deployment-matched student view and a stop-gradient privileged teacher view along the same generated trajectory, estimates the reward uplift of the privileged context, and uses the resulting sample-level weight to calibrate token-level distillation.

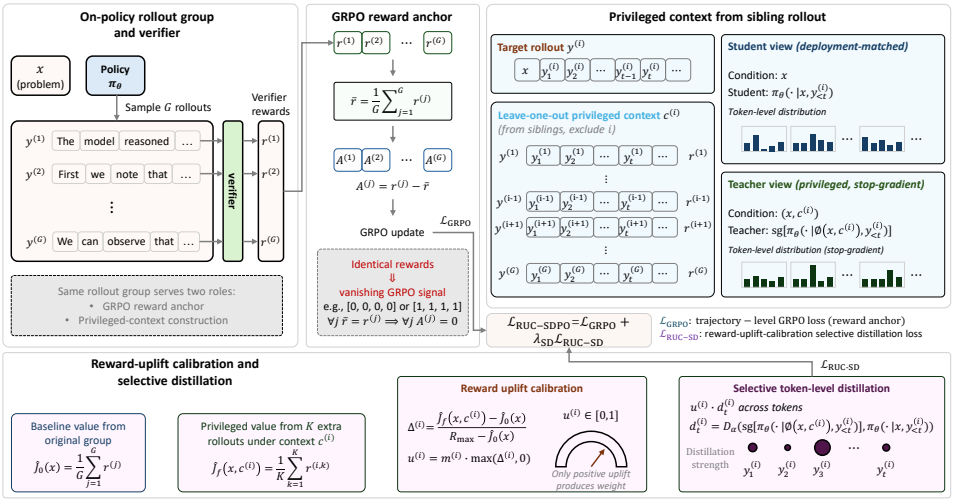


Figure 2: Overview of RUC-SDPO. On-policy rollouts and verifier outcomes provide both the GRPO reward anchor and the training-only privileged context for self-distillation. RUC-SDPO constructs a deployment-matched student view and a stop-gradient privileged teacher view of the same policy, estimates the reward uplift of the privileged context, and uses the resulting bounded weight to calibrate token-level distillation while preserving GRPO as the reward-anchored objective.

RUC-SDPO is designed as a reward-anchored augmentation of GRPO. When a sampled group contains useful reward contrast, the GRPO term remains the primary policy-gradient signal and directly optimizes verifier reward. The auxiliary self-distillation term dominates the update when the finite-sample GRPO estimator becomes weak or uninformative, such as the case when group rewards are homogeneous or all sampled trajectories fail. In these cases, a privileged teacher view with estimated positive reward uplift can provide dense token-level guidance while the deployment policy remains conditioned only on the original input. The following subsections define each component in detail.

3.1 Privileged Context Construction

RUC-SDPO uses the same on-policy rollouts for both reward-based policy optimization and privileged-context construction for self-distillation. Let π_θ denote the current policy with parameters θ. For an input x, we sample G trajectories $y^{(i)} \sim \pi_\theta(\cdot \mid x)$ and score each with the verifier R to obtain a reward $r^{(i)} = R(x, y^{(i)})$, which yields the rollout group $\mathcal{G} = \{(y^{(i)}, r^{(i)})\}_{i=1}^{G}$ of sampled trajectories and their verifier rewards. For each trajectory $y^{(i)}$, we construct a training-only privileged context $c^{(i)} = \mathcal{C}(x, \mathcal{G}_{-i})$ from its leave-one-out rollout group $\mathcal{G}_{-i} = \{(y^{(j)}, r^{(j)}) : j \neq i\}$, where $\mathcal{C}$ is a context-construction rule that forms the privileged context using information associated with the same on-policy rollout group, such as non-self sibling trajectories and their verifier outcomes. The privileged context is used only during training and is not available to the deployed policy. If no valid privileged context is available for a trajectory, we set $c^{(i)} = \varnothing$ and define a binary mask $m^{(i)} = \mathbf{1}[c^{(i)} \neq \varnothing]$, which will be used to deactivate the privileged distillation for these trajectories (detailed in the next subsection), such that they contribute to policy updates only through the GRPO reward objective.

Let $\phi(x, c^{(i)})$ denote the teacher input obtained by augmenting the original input x with the privileged context $c^{(i)}$. RUC-SDPO then evaluates two views of the policy along the same student-generated trajectory $y^{(i)}$, where $y^{(i)}_t$ denotes the t-th response token and $y^{(i)}_{<t}$ refers to its preceding tokens. The deployment-matched student view is conditioned only on the original input,

$$p^{(i)}_{S,t}(\cdot) \triangleq \pi_\theta(\cdot \mid x, y^{(i)}_{<t}), \tag{1}$$

whereas the privileged teacher view is conditioned on the augmented input,

$$p^{(i)}_{T,t}(\cdot) \triangleq \text{sg}\left[\pi_\theta(\cdot \mid \phi(x, c^{(i)}), y^{(i)}_{<t})\right], \tag{2}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator, the privileged teacher view thus supplies targets but receives no gradient. Both views are evaluated on the same trajectory $y^{(i)}$ and differ only in their conditioning context. This construction preserves the on-policy nature of the data while exposing the student to training-only information.

3.2 Reward-Uplift Calibration

We argue that privileged guidance should be trusted only to the extent that the privileged context improves task-level utility. Therefore, rather than applying self-distillation uniformly, RUC-SDPO calibrates the distillation signal toward reward-improving directions by estimating the reward uplift induced by the privileged context.

For an input x, we first estimate the reward value of the original deployment-matched context using the on-policy rollout group:

$$\widehat{J}_0(x) = \frac{1}{G}\sum_{i=1}^{G} r^{(i)}. \tag{3}$$

This quantity estimates the expected verifier reward under the original input context. For a trajectory with a valid privileged context $c^{(i)}$, we then estimate the reward value of the privileged teacher view by sampling K additional rollouts from the policy conditioned on the augmented input:

$$\tilde{y}^{(i,k)} \sim \pi_\theta(\cdot \mid \phi(x, c^{(i)})), \qquad k = 1, \ldots, K. \tag{4}$$

Averaging the verifier rewards of these K rollouts gives the privileged reward estimate

$$\widehat{J}_f(x, c^{(i)}) = \frac{1}{K}\sum_{k=1}^{K} R(x, \tilde{y}^{(i,k)}). \tag{5}$$

The estimate $\widehat{J}_f(x, c^{(i)})$ measures the expected verifier reward obtained when the same policy is conditioned on the augmented input, the difference $\widehat{J}_f(x, c^{(i)}) - \widehat{J}_0(x)$ thus measures the reward uplift attributable to the privileged context.

We further convert this reward uplift into a bounded sample-level distillation weight. Let verifier rewards be normalized to a bounded range with maximum attainable reward $R_{\max}$, we adopt binary correctness rewards with $R_{\max} = 1$ in our experiments. The normalized reward uplift is then defined as:

$$\Delta^{(i)} = \begin{cases} \frac{\widehat{J}_f(x,c^{(i)}) - \widehat{J}_0(x)}{R_{\max} - \widehat{J}_0(x)}, & R_{\max} - \widehat{J}_0(x) > \epsilon, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Here $\epsilon > 0$ is a small threshold that disables uplift normalization when the remaining reward margin is negligible, and the denominator normalizes the improvement by the remaining reward margin. The final reward-uplift weight is

$$u^{(i)} = m^{(i)} \cdot \max\Big(\Delta^{(i)}, 0\Big). \tag{7}$$

This ensures that trajectories without a valid privileged context receive zero weight. At the same time, privileged contexts that do not improve verifier reward are suppressed, while reward-improving contexts receive a distillation weight proportional to their normalized uplift. The weight $u^{(i)}$ is treated as a detached scalar when optimizing the final objective loss such that the reward-uplift estimates receive no gradient update. In practice, we average the privileged rewards over all valid contexts of the same input to obtain a prompt-level $\widehat{J}_f(x)$ used in Eq. (6), so that all valid targets of x share one weight that takes fractional values even with binary rewards and $K = 1$.

3.3 Reward-Anchored Optimization

RUC-SDPO combines the reward-anchored GRPO objective with a reward-uplift-calibrated self-distillation objective. For each student trajectory $y^{(i)}$, we measure the discrepancy between the privileged teacher view and the deployment-matched student view at every response token:

$$d_t^{(i)} = D_\alpha\big(p_{T,t}^{(i)}, p_{S,t}^{(i)}\big) = \alpha\,\mathrm{KL}\big(p_{T,t}^{(i)} \,\|\, q_{\alpha,t}^{(i)}\big) + (1-\alpha)\,\mathrm{KL}\big(p_{S,t}^{(i)} \,\|\, q_{\alpha,t}^{(i)}\big), \tag{8}$$

where D_α denotes an α-weighted Jensen–Shannon divergence with mixed distribution $q^{(i)}_{\alpha,t} = \alpha p^{(i)}_{T,t} + (1-\alpha)p^{(i)}_{S,t}$.

Because the privileged teacher distribution is detached, the gradient of $d^{(i)}_t$ flows only through the student, minimizing it thus pulls the student toward tokens that the teacher assigns higher probability instead of matching the two distributions symmetrically. Measuring both terms against the mixture $q^{(i)}_{\alpha,t}$ bounds this pull: when the student assigns near-zero probability to a token the teacher prefers, a plain Kullback–Leibler term would diverge, whereas $\mathrm{KL}(p^{(i)}_{T,t} \,\|\, q^{(i)}_{\alpha,t})$ stays finite because $q^{(i)}_{\alpha,t} \geq \alpha p^{(i)}_{T,t}$.

Let T_i denote the number of response tokens in $y^{(i)}$. The reward-uplift-calibrated self-distillation loss is

$$\mathcal{L}_{\mathrm{RUC-SD}} = \frac{1}{\sum_i T_i} \sum_i \sum_{t=1}^{T_i} u^{(i)} d^{(i)}_t. \tag{9}$$

The sample-level weight $u^{(i)}$ is shared across all tokens of trajectory $y^{(i)}$, while the divergence itself provides token-level guidance. Therefore, privileged distillation is active only when the privileged context yields positive reward uplift, and its strength is proportional to the estimated task-level benefit. RUC-SDPO thus combines trajectory-level reward calibration with dense token-level distillation: reward uplift decides whether the privileged teacher view should be trusted for the sample, but it does not attribute reward to individual tokens, and the token-level targets come from the divergence.

The final optimization objective is

$$\mathcal{L}_{\mathrm{RUC-SDPO}} = \mathcal{L}_{\mathrm{GRPO}} + \lambda_{\mathrm{SD}} \mathcal{L}_{\mathrm{RUC-SD}}, \tag{10}$$

where λ_{SD} controls the strength of the auxiliary distillation term, and $\mathcal{L}_{\mathrm{GRPO}}$ denotes the strengthened GRPO objective defined in Appendix A. GRPO remains the primary reward anchor, directly optimizing the policy using verifier-derived trajectory rewards, while the calibrated self-distillation term provides dense token-level guidance only for reward-improving privileged contexts. This design avoids blindly matching the privileged teacher distribution and reduces the risk of distilling context-specific preferences that are not supported by task-level reward improvement. Algorithm 1 in Appendix E summarizes the complete training procedure as implemented in our experiments.

4 Experiments

4.1 Experimental Setup

Training data and evaluation benchmarks. We evaluate RUC-SDPO on two task families: mathematical reasoning and tool-call generation (Schick et al., 2023). For mathematical reasoning, we train on OpenR1-Math (Open-R1 Team, 2025) and evaluate on external benchmarks, including MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023), AIME 2026 (Mathematical Association of America, 2026), and HMMT February 2026 (Harvard-MIT Mathematics Tournament, 2026). For tool-call generation, we train on ToolACE (Liu et al., 2025a) and evaluate on ACEBench (Chen et al., 2025) and the Berkeley Function-Calling Leaderboard (BFCL) (Patil et al., 2025). For BFCL, we follow the official single-turn Abstract Syntax Tree (AST) evaluation protocol and report results on the Non-Live and Live subsets. No evaluation benchmark is used for training.

Compared methods. We compare RUC-SDPO against three baselines. Base refers to the initial policy before task-specific training. Self-Distillation Policy Optimization (SDPO) (Hübotter et al., 2026) performs on-policy self-distillation using the same privileged-context construction as RUC-SDPO, but without reward-uplift calibration. GRPO (Shao et al., 2024) denotes our strengthened GRPO baseline, which uses DAPO-style (Yu et al., 2025) asymmetric clipping, applies token-level importance-sampling correction for rollout-policy mismatch (Khatri et al., 2025), and disables reward standard-deviation normalization in the group-relative advantage estimate (Liu et al., 2025b). RUC-SDPO is our full method, which adds reward-uplift-calibrated self-distillation to the GRPO reward anchor.

Training setup. For each task family, we convert the official training data into a unified format. Each converted sample consists of a prompt and a reference output, where the reference output is used by the verifier to compute rewards. Unless otherwise specified, each task-specific training run uses $9{,}600$ samples from the converted official training data. Policy checkpoints are saved every 10 optimization steps. Appendix B details data processing, task formats, verifiers, and dataset statistics.

All task-specific training runs are conducted using eight NVIDIA H20 GPUs with the same training data size, number of epochs, and training rollout decoding configuration across the compared training methods. For each prompt, we sample $G = 8$ responses during training. Unless otherwise specified, each run uses a learning rate of 1×10^{-6} and one training epoch. RUC-SDPO uses $\lambda_{\mathrm{SD}} = 0.01$, and sets $\alpha = 0.5$ for the Jensen–Shannon divergence, and estimates reward uplift with $K = 1$ additional rollout for each constructed privileged context. Additional implementation details and task-specific hyperparameters are provided in Appendix C.

RUC-SDPO constructs privileged contexts from sibling responses sampled for the same prompt. To construct the privileged context $c^{(i)}$ for a rollout $y^{(i)}$, we randomly select one response from the remaining $G - 1$ responses sampled for the same prompt, remove any intermediate reasoning traces, and include it together with its verifier outcome. The constructed context is used only as training-time privileged information, while the student-side conditioning remains the original prompt. Appendix D provides the exact context-construction template and details on prompt-group-level uplift aggregation.

Evaluation protocol. For trained methods, we evaluate the final checkpoint on the external benchmarks corresponding to the same task family; for Base, we evaluate the initial policy directly without task-specific training. For mathematical reasoning benchmarks, each evaluation samples 8 responses per prompt and reports both MEAN@8 and BEST@8. We repeat mathematical evaluation with three random seeds and report the mean and standard deviation. MEAN@k measures the average verifier score across k sampled responses,

$$\text{MEAN@}k = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \frac{1}{k} \sum_{i=1}^{k} R(x, y^{(i)}), \tag{11}$$

while BEST@k measures whether any of the sampled responses solves the prompt,

$$\text{BEST@}k = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \max_{1 \le i \le k} R(x, y^{(i)}). \tag{12}$$

Here $\mathcal{D}$ denotes the set of benchmark prompts.

For tool-call evaluation, we report the ACEBench overall score and the official BFCL single-turn AST results on the Non-Live and Live subsets. Appendix C provides decoding settings for all external evaluations.

4.2 Main Results

Table 1 shows that RUC-SDPO consistently improves mathematical MEAN@8 over the strengthened GRPO baseline across all three model scales and all three mathematical benchmarks. RUC-SDPO attains the best MEAN@8 on MATH-500, AIME 2026, and HMMT February 2026 for Qwen3-4B, Qwen3-8B, and Qwen3-14B, indicating more reliable per-sample solution quality rather than gains that rely only on larger sampling budgets. The MEAN@8 gains range from 0.78 to 2.64 points across the nine model–benchmark settings, with larger gains on AIME 2026 and HMMT February 2026 than the more saturated MATH-500 benchmark. RUC-SDPO also improves most BEST@8 results over GRPO. The main exception is MATH-500 BEST@8 on Qwen3-4B, where GRPO remains slightly higher (91.33 ± 0.50 vs. 91.07 ± 0.12); on this saturated benchmark, RUC-SDPO still delivers the best MEAN@8.

For tool-call generation, RUC-SDPO obtains the best score in eight of nine model–benchmark settings. It obtains the best results on ACEBench and BFCL Non-Live AST for all three model scales, and also reaches the best BFCL Live AST score on Qwen3-8B and Qwen3-14B. The only tool-call exception is BFCL Live AST on Qwen3-4B, where RUC-SDPO is narrowly second to GRPO (81.94 vs. 82.01). The uncalibrated SDPO baseline improves over Base in many settings but

Table 1: Main results across the two task families. Mathematical reasoning checkpoints are evaluated on MATH-500, AIME 2026, and HMMT February 2026 with MEAN@8 and BEST@8, reported as mean and standard deviation across repeated evaluations; tool-call generation checkpoints are evaluated on ACEBench and the BFCL Non-Live and Live AST subsets. Within each model block, the best result is shown in bold and the second-best result is underlined.

Method	Mathematical reasoning						Tool-call generation		
	MATH-500		AIME 2026		HMMT Feb. 2026		ACEBench	BFCL Non-Live	BFCL Live
	MEAN@8↑	BEST@8↑	MEAN@8↑	BEST@8↑	MEAN@8↑	BEST@8↑	overall↑	AST score↑	AST score↑
Qwen3-4B									
Base	75.81±0.28	86.53±0.42	14.44±0.96	32.22±1.92	14.52±1.53	30.30±3.03	45.14	86.77	80.46
SDPO	80.00±0.50	89.40±0.40	25.83±0.83	52.22±1.92	21.84±0.95	35.35±1.75	55.14	<u>89.40</u>	81.20
GRPO	<u>85.44±0.12</u>	**91.33±0.50**	<u>40.83±1.25</u>	<u>68.89±1.92</u>	<u>27.90±0.87</u>	<u>39.39±0.00</u>	<u>61.14</u>	88.69	**82.01**
RUC-SDPO	**86.22±0.20**	<u>91.07±0.12</u>	**42.50±0.83**	**71.11±1.92**	**29.55±0.76**	**41.41±1.75**	**62.00**	**89.92**	<u>81.94</u>
Qwen3-8B									
Base	76.44±0.30	87.33±0.31	16.53±0.64	34.44±1.92	15.53±1.52	33.33±3.03	46.86	88.81	81.05
SDPO	82.50±0.50	90.00±0.40	30.00±0.83	56.67±3.33	23.48±0.76	37.37±1.75	55.43	<u>89.80</u>	81.80
GRPO	<u>85.94±0.29</u>	<u>92.13±0.42</u>	<u>45.69±1.25</u>	<u>71.11±3.85</u>	<u>28.28±0.79</u>	<u>38.38±1.75</u>	<u>55.71</u>	89.40	<u>82.50</u>
RUC-SDPO	**87.03±0.16**	**92.80±0.20**	**48.33±0.83**	**74.44±1.92**	**30.68±0.76**	**42.42±0.00**	**58.00**	**90.50**	**83.80**
Qwen3-14B									
Base	79.19±0.54	88.47±0.12	20.14±0.64	41.11±1.92	19.70±0.38	37.37±3.50	60.00	89.23	82.09
SDPO	84.50±0.50	90.20±0.20	34.17±0.83	63.33±3.33	27.27±0.76	40.40±1.75	66.00	89.90	82.60
GRPO	<u>86.83±0.25</u>	<u>90.60±0.20</u>	<u>47.22±1.68</u>	<u>70.00±0.00</u>	<u>31.94±0.44</u>	<u>43.43±3.50</u>	<u>70.57</u>	<u>90.56</u>	<u>83.20</u>
RUC-SDPO	**87.88±0.12**	**91.40±0.20**	**49.58±0.83**	**72.22±1.92**	**33.46±0.58**	**44.44±1.75**	**72.57**	**91.20**	**84.10**

Table 2: Effect of the GRPO reward anchor on Qwen3-8B. RUC-SD denotes reward-uplift-calibrated self-distillation without the GRPO reward objective. Avg. denotes the average over the three mathematical benchmarks using MEAN@8; Δ and Wins compare each GRPO-anchored variant with the method in the row immediately above it, where Wins counts the benchmarks on which the variant scores higher.

Method	MATH	AIME	HMMT	Avg.	Δ	Wins
SDPO	82.5±0.5	30.0±0.8	23.5±0.8	45.3	–	–
GRPO+SDPO	84.1±0.4	36.4±0.6	26.0±0.6	48.8	+3.5	3/3
RUC-SD	85.7±0.4	42.8±0.6	28.5±0.6	52.3	–	–
RUC-SDPO	87.0±0.2	48.3±0.8	30.7±0.8	55.3	+3.0	3/3

Table 3: Effect of reward-uplift calibration under the same GRPO reward anchor on Qwen3-8B. GRPO+SDPO uses an uncalibrated distillation weight, GRPO+Binary Gate activates distillation only under positive estimated uplift, and RUC-SDPO uses the normalized clipped reward-uplift weight. Avg. denotes the average over the three mathematical benchmarks using MEAN@8; Δ and Wins compare each distillation variant against GRPO.

Method	MATH	AIME	HMMT	Avg.	Δ	Wins
GRPO	85.9±0.3	45.7±1.3	28.3±0.8	53.3	–	–
GRPO+SDPO	84.1±0.4	36.4±0.6	26.0±0.6	48.8	−4.5	0/3
GRPO +Binary Gate	86.4±0.5	46.8±0.5	29.4±0.6	54.2	+0.9	3/3
RUC-SDPO	87.0±0.2	48.3±0.8	30.7±0.8	55.3	+2.0	3/3

remains consistently inferior to RUC-SDPO, supporting the central design of RUC-SDPO: GRPO provides the reward-grounded anchor, while reward-uplift-calibrated privileged distillation adds dense guidance when the privileged teacher view is useful.

4.3 Ablation Studies

We ablate RUC-SDPO on Qwen3-8B mathematical reasoning, where verifier rewards are sparse and the interaction between reward anchoring and dense distillation is most pronounced. The ablations isolate the two design choices introduced by the method: retaining GRPO as a reward anchor and calibrating privileged self-distillation by estimated reward uplift. Unless otherwise specified, ablation variants use the same training data, rollout budget, and evaluation protocol as the main experiments. We report the average MEAN@8 over the three mathematical benchmarks and count the number of settings in which each variant improves over its corresponding baseline.

We first examine whether the dense distillation signal should replace or complement reward optimization. As shown in Table 2, adding the GRPO reward anchor improves both the uncalibrated and reward-uplift-calibrated distillation variants, with gains on all three benchmarks. Meanwhile, RUC-

Table 4: Selective behavior of reward-uplift calibration on Qwen3-8B. Pos. uplift is the fraction of valid privileged contexts with positive estimated uplift, Suppressed is the fraction assigned zero distillation weight, and Avg. weight is the average detached weight.

Group subset	Pos. uplift	Suppressed	Avg. weight
All valid groups	48.0%	52.0%	0.272
Identical-reward groups	58.2%	41.8%	0.348
Mixed-reward groups	23.8%	76.2%	0.092

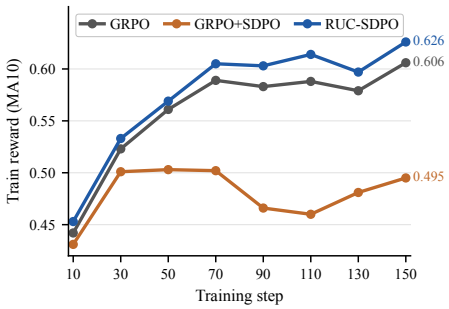


Figure 3: Training dynamics on Qwen3-8B mathematical reasoning. MA10 denotes a 10-step moving average. The labels at the right edge report the final moving-average reward at step 150.

SD substantially outperforms SDPO before adding the GRPO anchor, indicating that reward-uplift calibration is useful even as a standalone distillation objective.

We next isolate the effect of reward-uplift calibration by keeping the GRPO reward anchor fixed and varying only how the auxiliary distillation loss is weighted. As shown in Table 3, the uncalibrated variant underperforms GRPO, showing that dense distillation can be harmful when applied indiscriminately. Binary gating recovers the lost performance by suppressing non-positive uplift cases, while our continuous reward-uplift weighting improves all three benchmarks.

4.4 Mechanism and Cost Analysis

As a mechanism check, we analyze calibration weights across reward regimes. Table 4 shows that 48.0% of valid privileged contexts receive positive uplift, whereas the remaining 52.0% receive zero distillation weight, indicating that distillation is not universally activated. Among valid privileged contexts in identical-reward groups, where GRPO's group-relative signal vanishes, 58.2% receive positive uplift, with an average weight of 0.348. The corresponding figures for mixed-reward groups are 23.8% and 0.092, respectively. This contrast is consistent with the intended role of RUC-SDPO: providing dense guidance when GRPO lacks a relative reward signal while suppressing distillation when the estimated uplift is non-positive.

The training dynamics in Figure 3 further align with these endpoint ablations. RUC-SDPO maintains higher moving-average training rewards throughout training, whereas uncalibrated GRPO+SDPO falls below GRPO after early updates. This pattern supports the role of reward-uplift calibration as a filter on the auxiliary distillation signal rather than merely an additional loss term.

Because RUC-SDPO uses additional privileged-context rollouts, its average per-step training time increases from 292 seconds for GRPO to 371 seconds, an overhead of about 27%.

5 Conclusion

We introduced RUC-SDPO, a reward-anchored extension of GRPO that calibrates privileged on-policy self-distillation by estimated verifier-reward uplift. The method addresses coarse or vanishing group-relative learning signals by adding dense token-level guidance only when privileged conditioning is estimated to improve task reward. By mapping positive uplift to a bounded detached weight, RUC-SDPO keeps distillation auxiliary, selective, and tied to verifier outcomes without requiring a separate teacher model, learned reward model, or step-level annotations. Experiments on mathematical reasoning and tool-call generation show consistent gains over strengthened GRPO baselines, while ablations and training-dynamics analysis support the roles of the GRPO reward anchor and reward-uplift calibration. These results suggest that reward-uplift calibration is an effective way to turn training-only privileged context into reward-grounded dense guidance for RLVR.

AI USE STATEMENT

Large language models assist with editorial polishing of the manuscript. The authors determine the scientific claims, supply the experimental results, check the mathematical content, and remain responsible for the final text.

REFERENCES

Chen Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Yuefeng Huang, Xiangcheng Liu, Xinzhi Wang, and Wu Liu. ACEBench: A comprehensive evaluation of LLM tool usage. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 12970–12998, 2025. doi: 10.18653/v1/2025.findings-emnlp.697. URL https://aclanthology.org/2025.findings-emnlp.697/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ken Ding. HDPO: Hybrid distillation policy optimization via privileged self-distillation. *arXiv preprint arXiv:2603.23871*, 2026.

Harvard-MIT Mathematics Tournament. HMMT february 2026. https://www.hmmt.org/www/archive/292, 2026.

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

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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

Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Sai Surya Duvvuri, Manzil Zaheer, Inderjit S. Dhillon, David Brandfonbrener, and Rishabh Agarwal. The art of scaling reinforcement learning compute for LLMs. *arXiv preprint arXiv:2510.13786*, 2025.

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

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Xinzhi Wang, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. ToolACE: Winning the points of LLM function calling. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=8EB8k6DdCU.

Yifeng Liu, Shiyuan Zhang, Yifan Zhang, and Quanquan Gu. Self-distilled policy gradient. *arXiv preprint arXiv:2606.04036*, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b.

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

Mathematical Association of America. American invitational mathematics examination (AIME) 2026. `https://maa.org/maa-invitational-competitions/`, 2026.

Open-R1 Team. OpenR1-Math-220K: A large-scale dataset for mathematical reasoning. `https://huggingface.co/datasets/open-r1/OpenR1-Math-220k`, 2025.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guobin Shen, Lei Huang, Xiang Cheng, Chenxiao Zhao, Jindong Li, Dongcheng Zhao, and Xing Yu. From generic correlation to input-specific credit in on-policy self distillation. *arXiv preprint arXiv:2605.11613*, 2026.

Alex Stein, Furong Huang, and Tom Goldstein. Gates: Self-distillation under privileged context with consensus gating. *arXiv preprint arXiv:2602.20574*, 2026.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. In *The Eleventh International Conference on Learning Representations*, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488, 2022.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Binbin Zheng, Xing Ma, Yiheng Liang, Jingqing Ruan, Xiaoliang Fu, Kepeng Lin, Benchang Zhu, Ke Zeng, and Xunliang Cai. Scope: Signal-calibrated on-policy distillation enhancement with dual-path adaptive weighting. *arXiv preprint arXiv:2604.10688*, 2026.

A STRENGTHENED GRPO OBJECTIVE

This appendix defines the strengthened GRPO objective $\mathcal{L}_{\mathrm{GRPO}}$ used as the reward anchor in RUC-SDPO and as the GRPO baseline in our experiments. The construction follows GRPO (Shao et al., 2024) with three standard refinements from prior work: DAPO-style asymmetric clipping (Yu et al., 2025), token-level importance-sampling correction for rollout-policy mismatch (Khatri et al., 2025), and group-relative advantages without reward standard-deviation normalization (Liu et al., 2025b). These refinements are not claimed as contributions of this paper.

Given a rollout group $\mathcal{G} = \{(y^{(i)}, r^{(i)})\}_{i=1}^{G}$, define the mean-centered group-relative advantage

$$A^{(i)} = r^{(i)} - \frac{1}{G}\sum_{j=1}^{G} r^{(j)}. \tag{13}$$

Let $\pi_{\theta_{\mathrm{old}}}$ denote the rollout policy that generated $y^{(i)}$, and define the token-level importance ratio

$$\rho_t^{(i)} = \frac{\pi_\theta\left(y_t^{(i)} \mid x, y_{<t}^{(i)}\right)}{\pi_{\theta_{\mathrm{old}}}\left(y_t^{(i)} \mid x, y_{<t}^{(i)}\right)}. \tag{14}$$

Following (Khatri et al., 2025), we apply a truncated token-level importance weight

$$w_t^{(i)} = \min\left(\rho_t^{(i)},\, C\right), \tag{15}$$

where C is a truncation threshold. With DAPO-style asymmetric clipping bounds $\varepsilon_{\mathrm{low}}$ and $\varepsilon_{\mathrm{high}}$, the per-token surrogate is

$$s_t^{(i)} = \min\Big(w_t^{(i)} A^{(i)},\, \mathrm{clip}\big(w_t^{(i)},\, 1-\varepsilon_{\mathrm{low}},\, 1+\varepsilon_{\mathrm{high}}\big) A^{(i)}\Big). \tag{16}$$

Let T_i denote the number of response tokens in $y^{(i)}$. The strengthened GRPO loss used in the main text is

$$\mathcal{L}_{\mathrm{GRPO}} = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i} s_t^{(i)}. \tag{17}$$

Hyperparameter values for $\varepsilon_{\mathrm{low}}$, $\varepsilon_{\mathrm{high}}$, and C are reported in Appendix C.

B DATA PROCESSING AND TASK DETAILS

This appendix describes the data processing pipeline for the official training datasets used in our experiments. We use OpenR1-Math for mathematical reasoning and ToolACE for tool-call generation. Raw data are converted into a unified format in which each sample contains a prompt and a verifier-compatible reference output. The reference output is used only for reward computation and is never inserted into the policy prompt during rollout generation. RUC-SDPO does not use ground-truth rationales or reference solutions as privileged context; privileged contexts are constructed online from sibling policy rollouts within the same prompt group. Final evaluation is conducted only on the external benchmarks described in the Experimental Setup section.

B.1 UNIFIED DATA SCHEMA

Table 5 summarizes the conceptual schema shared by converted training samples.

B.2 DATASET SOURCES AND STATISTICS

Table 6 summarizes the processed training datasets. For OpenR1-Math, we remove duplicate problems from the 225,129 raw samples, resulting in 190,302 processed samples. For ToolACE, we first apply item-level filtering to remove dialogues with invalid system tool specifications or no retainable assistant tool-call turns, leaving 9,150 items from 11,300 raw items. These items are further decomposed into 9,853 tool-call samples: 8,514 items yield one sample, 571 yield two samples, 63 yield three samples, and 2 yield four samples.

Table 5: Unified schema for converted training samples.

Field	Description
Prompt	The task prompt presented to the policy.
Reference output	The verifier-compatible reference used for reward computation.
Source	The official dataset from which the sample is converted.

Table 6: Statistics of the converted official training datasets.

Task family	Source	Raw	Proc.	Notes
Math	OpenR1-Math	225,129	190,302	Duplicate problems removed.
Tool-call	ToolACE	11,300	9,853	Filtered into tool-call samples.

B.3 Task Details

OpenR1-Math. Each sample contains a mathematical problem. The prompt instructs the model to solve the problem, show its reasoning, and put the final answer in a boxed expression. The reference output is the final mathematical answer. The verifier extracts the model's final boxed answer and checks answer equivalence against the reference answer. Responses without a valid final boxed answer are treated as format errors and receive zero reward.

ToolACE. Each sample is derived from a tool-calling dialogue. The model is given the dialogue context before an assistant tool-call turn, together with the available tool definitions. The prompt renders the conversation history and asks the model to produce the next tool call for the latest user request. The reference output is the assistant tool call from the original dialogue, canonicalized into the format used by the verifier.

B.4 Tool-Call Canonicalization

For ToolACE, both predicted and reference tool calls are converted into a canonical representation before scoring. Each call is represented by the tool name and a JSON object of arguments. Argument objects are serialized with deterministic key ordering, and invalid JSON action inputs are treated as malformed predictions. A prediction is marked correct only when the parsed tool-call structure matches the canonical reference structure.

C Training Hyperparameters

This appendix summarizes the hyperparameters used in our main experiments. Table 7 reports the common training settings shared by GRPO, SDPO, and RUC-SDPO unless otherwise specified. Table 8 lists RUC-SDPO-specific implementation settings, and Table 9 reports external evaluation decoding settings.

For BFCL evaluation, the generation budget follows the official handler: $M_{\text{gen}} = \min(4096, L_{\text{ctx}} - L_{\text{in}} - 2)$, where L_{ctx} is the maximum context length and L_{in} is the input token count. The vLLM service uses $L_{\text{ctx}} = 40960$.

The EMA policy is not a separately trained teacher model; it is an exponential moving average of the current policy, and is used to stabilize privileged-view targets and reward-uplift rollouts. This implementation choice does not change the deployment-time policy, which remains conditioned only on the original prompt. For implementation stability, we aggregate reward-uplift estimates at the prompt-group level: all valid self-distillation targets derived from the same original prompt share the same detached uplift weight. This aggregation affects only the scalar weighting of the auxiliary distillation loss and leaves the GRPO reward objective unchanged.

Table 7: Common hyperparameters used in the main experiments.

Config	Value	Description
Training samples	9,600	Number of training samples used for each run.
Train batch size	64	Number of prompts in each training batch.
PPO mini-batch size	64	Mini-batch size for actor update iterations.
Micro-batch size	1	Per-GPU micro-batch size for actor and log-probability computation.
Training rollouts	8	Number of rollout responses sampled per prompt during training.
Learning rate	1×10^{-6}	Actor learning rate.
Warmup steps	10	Number of learning-rate warmup steps.
Training epochs	1	Number of passes over the training data.
Save frequency	10	Checkpointing frequency in optimization steps.
Prompt / response length	2,048 / 8,192	Maximum prompt and response lengths.
Model context length	18,944	Maximum context length used by rollout and actor updates.
Training sampling	1.0 / 1.0 / -1	Temperature / top-p / top-k for rollout generation.
Asymmetric clip range	0.2 / 0.28	Lower / upper clipping bounds for the PPO/GRPO policy ratio.
Rollout correction	token / 2.0	Token-level importance-sampling correction and truncation threshold.

Table 8: RUC-SDPO-specific hyperparameters and implementation settings used in the main experiments.

Config	Value	Description
Divergence	JSD	Token-level self-distillation divergence.
JSD coefficient α	0.5	Interpolation coefficient for the Jensen–Shannon divergence.
EMA policy	update rate 0.05	Exponential moving average of the current policy, used only as an implementation stabilizer for privileged-view target distributions and reward-uplift rollouts.
$\widehat{J}_f$ rollout policy	EMA policy	Privileged-context rollouts for estimating $\widehat{J}_f$ are sampled from the EMA policy for stability.
Uplift aggregation	prompt-group level	Valid distillation targets derived from the same original prompt share the same detached uplift weight.
Calibration rollouts K	1	Additional privileged-context rollout per valid sample.
Reward upper bound	1.0	Maximum verifier reward used for uplift normalization.
Distillation top-k	100	Number of logits retained in the distillation approximation.
λ_{SD}	0.01	Coefficient for the auxiliary self-distillation loss.
Sibling demonstration sampling	random non-self	Demonstration is sampled from non-self sibling responses.
Distillation IS clip	2.0	Importance-ratio clip used in distillation.

D Privileged Context Construction

This appendix describes the prompt template used to instantiate the privileged context in our experiments. We use the notation from the main text: for the i-th rollout, $c^{(i)}$ denotes the training-only privileged context and $\phi(x, c^{(i)})$ denotes the augmented teacher input. The student input remains the original task prompt x.

D.1 Prompt Template

In the main experiments, $c^{(i)}$ is constructed from a randomly selected non-self sibling response in the same rollout group. The selected sibling can be either correct or incorrect according to the verifier.

Table 9: Decoding settings used for external benchmark evaluation.

Config	Mathematical reasoning	ACEBench	BFCL
Benchmarks	MATH-500, AIME 2026, HMMT February 2026	ACEBench	BFCL Non-Live and Live
Samples per prompt	8; reported as MEAN@8 and BEST@8	1	1
Temperature	0.6	0.0	≈ 0.001
Top-p	0.95	1.0	1.0
Top-k	-1	–	–
Maximum tokens	32,768	2,048	dynamic, capped at 4,096
Seeds	42 / 43 / 44	deterministic	deterministic
Thinking mode	disabled	–	–

Before the sibling response is inserted into the teacher input, intermediate reasoning traces enclosed by `<think>...</think>` are removed when present.

The augmented teacher input $\phi(x, c^{(i)})$ is formed by appending an outcome-labeled demonstration block to the original prompt. The templates are:

Correct sibling demonstration

Correct solution:
<selected sibling response>
Correctly solve the original question.

Incorrect sibling demonstration

Incorrect solution:
<selected sibling response>
Correctly solve the original question.

The outcome label is part of the privileged context and indicates whether the sibling attempt should be treated as a successful or unsuccessful demonstration. The label and sibling response are used only in the privileged teacher view during training and are never included in the student input or deployment-time prompt.

D.2 Reward-Uplift Aggregation

The Reward-Uplift Calibration subsection in the main text defines the reward-uplift estimate $\widehat{J}_f(x, c^{(i)})$ for a privileged context. In implementation, multiple target rollouts from the same prompt can yield related augmented teacher inputs. We therefore aggregate reward-uplift estimates at the prompt-group level in the main experiments. For a fixed original prompt x, we pool the teacher-conditioned rewards from all valid privileged contexts constructed within the same rollout group and use their average as the prompt-level estimate of $\widehat{J}_f$. All valid self-distillation targets derived from the same original prompt then share the same detached uplift weight. This reduces variance when verifier rewards are sparse or binary.

An alternative is privileged-context-level aggregation, which keeps separate estimates for distinct augmented teacher inputs $\phi(x, c^{(i)})$. This variant can distinguish different sibling demonstrations for the same original prompt, but is noisier when the number of teacher-conditioned calibration rollouts K is small. It is not used in the main experiments. The main experiments use prompt-group-level aggregation with $K = 1$ additional EMA-policy rollout per valid privileged context.

E Training Procedure

Algorithm 1 summarizes one training step of RUC-SDPO as implemented in the main experiments, including the sibling-based context construction, the EMA policy, and prompt-group-level uplift aggregation described in Appendices C and D.

Algorithm 1 Reward-Uplift-Calibrated Self-Distillation Policy Optimization

Require: Policy π_θ, EMA policy $\pi_{\bar{\theta}}$ with update rate η, dataset $\mathcal{D}$, verifier R, group size G, calibration rollouts K, coefficient λ_{SD}

for each training step **do**
  for each prompt x in a batch sampled from $\mathcal{D}$ **do**
    Sample $y^{(i)} \sim \pi_\theta(\cdot \mid x)$ and compute $r^{(i)} = R(x, y^{(i)})$ for $i = 1, \ldots, G$.
    Compute group-relative advantages and $\mathcal{L}_{\mathrm{GRPO}}$ (Appendix A).
    Estimate $\widehat{J}_0(x)$ by Eq. (3).
    for $i = 1, \ldots, G$ **do**
      Select a random sibling $y^{(j)}$ with $j \neq i$, remove its reasoning trace, and form $c^{(i)}$ with the outcome label of $r^{(j)}$ (Appendix D); set $m^{(i)}$.
      If $m^{(i)} = 1$, sample $\tilde{y}^{(i,k)} \sim \pi_{\bar{\theta}}(\cdot \mid \phi(x, c^{(i)}))$ for $k = 1, \ldots, K$.
    end for
    Pool $R(x, \tilde{y}^{(i,k)})$ over all valid contexts and all k into the prompt-level estimate $\widehat{J}_f(x)$.
    Compute $\Delta(x)$ by Eq. (6) with $\widehat{J}_f(x)$, and set the detached weight $u^{(i)} = m^{(i)} \max(\Delta(x), 0)$ as in Eq. (7).
    Evaluate $p^{(i)}_{S,t}$ under π_θ and the detached targets $p^{(i)}_{T,t}$ under $\pi_{\bar{\theta}}$ by Eqs. (1)–(2).
    Compute $\mathcal{L}_{\mathrm{RUC-SD}}$ by Eqs. (8)–(9).
  end for
  Update π_θ by minimizing Eq. (10).
  Update the EMA parameters $\bar{\theta} \leftarrow (1 - \eta)\bar{\theta} + \eta\theta$.
end for