GROUP-MARGINALIZED SELF-REWARDING RL DRIVES ZERO-LABEL SELF-EVOLVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-rewarding reinforcement learning (RL) enables large language models (LLMs) to self-evolve without human labels. Existing ensemble-based methods construct reward references from rollout groups and assign rewards accordingly. However, a response's reward representation also depends on its randomly sampled group context, *i.e.*, the other responses in its group. Using only one group-context realization may miss desired reward signals and provide unreliable guidance for policy optimization. To address this issue, we propose **Group-Marginalized Advantage Estimation (GMAE)**, which aggregates reward realizations across possible contexts into a response-level distribution and estimates expected advantages. Experiments across eight benchmarks and four base models demonstrate strong performance and cross-domain generalization. GMAE also exhibits stable learning, low extra cost, and good applicability across training datasets and RL backbones.

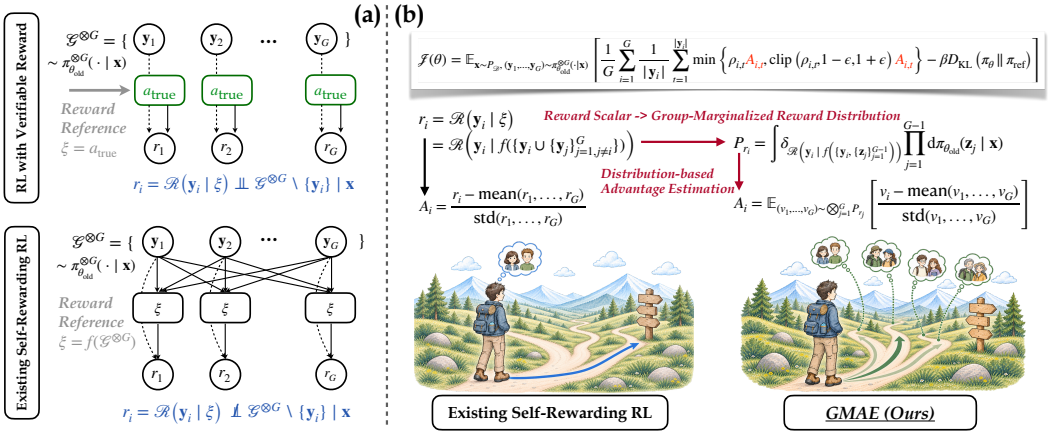


Figure 1: **(a) GMAE Motivation:** RLVR evaluates each response against external ground truth, whereas in self-rewarding RL its reward representation depends on the remaining responses in its rollout group, termed its *group context*. **(b) GMAE Overview:** Conventional methods use a single group-context realization to obtain each response's reward representation, while GMAE aggregates rewards across possible contexts into a reward distribution and estimates the expected advantage.

1 INTRODUCTION

Self-evolution enables large language models (LLMs) to learn autonomously from their own experience (Tao et al., 2024), yet becomes harder as models increasingly decide what, when, and how to evolve (Gao et al., 2025). A fundamental setting is *zero-label* learning, where models derive signals from environments and data without human labels. In RL post-training, this motivates *self-rewarding reinforcement learning* (Yuan et al., 2024; Zhao et al., 2026b), which uses self-generated rewards for policy optimization, unlike *RL with verifiable rewards (RLVR)* relying on ground-truth labels.

Existing self-rewarding RL methods mainly derive supervision from two sources of self-generated information: probability-based and ensemble-based signals (He et al., 2026). Probability-based methods estimate response quality from token probabilities or entropy (Zhao et al., 2026b; Li et al., 2025), but often provide weak supervision and may lead to training instability or collapse (Zhang et al., 2025; Roy et al., 2025). Ensemble-based methods instead aggregate the rollout response group into a reward reference, typically a pseudo-label obtained through majority voting (Wang et al.,

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.

2023a; Zuo et al., 2026), and reward each response against this reference. Owing to its simplicity and effectiveness, this paradigm is widely adopted, with subsequent methods refining reward-reference construction and reward function (Zhang et al., 2026a; Wang et al., 2026a; Wu et al., 2026).

Despite their promising performance, these ensemble-based methods share a hidden rewarding limitation, as illustrated in Figure 1(a). For a response used for policy optimization, its reward representation depends not only on the response itself but also on the other responses in its rollout group. These responses constitute its randomly sampled **group context**. This dependence on a random factor introduces uncertainty into the reward representation: a fixed response may receive the desired reward signal only under certain group contexts, as shown in Figure 2. However, existing methods represent this context-dependent reward using only a single group-context realization. If the sampled realization misses the desired signal, that signal cannot guide subsequent policy optimization. *Thus, relying on a single group-context realization for reward representation makes policy updates sensitive to random rollout sampling, reducing policy optimization reliability.*

Following the above analysis, a reward representation restricted to a single group-context realization captures the desired signal only when that realization happens to be suitable for the response. Since this depends on both the self-rewarding rule and the response characteristics, it cannot be guaranteed in practice. Therefore, we consider preserving more reward realizations across different group contexts to reduce the risk of missing desired signals. This motivates our **Group-Marginalized Advantage Estimation (GMAE)** strategy, illustrated in Figure 1(b). GMAE aggregates these reward realizations into a distribution that serves as the response-level reward representation, then uses it to estimate the response's expected advantage. Notably, GMAE is *a general strategy* agnostic to reward-reference construction and reward function: these components define the self-rewarding rules and serve as the "fuel", while GMAE acts as a general "vehicle" for leveraging them. To apply GMAE in practice, we instantiate it with three representative self-rewarding rules.

We conduct extensive experiments with four base models across eight benchmarks covering mathematics, science, knowledge, and coding. GMAE achieves strong performance and generalizes from mathematical training data to unseen domains, demonstrating more reliable reward estimation. It also exhibits stable learning dynamics and low variance across random seeds, demonstrating more stable optimization. Further experiments also confirm its applicability across training datasets and RL backbones. Overall, these results establish GMAE as a reliable general framework for zero-label self-evolution through self-rewarding RL.

2 Preliminaries: RLVR and Self-Rewarding RL

In modern LLM post-training, RLVR is a common RL paradigm that improves reasoning through verifiable rewards. As a representative algorithm, GRPO (Shao et al., 2024) uses a PPO-style objective (Schulman et al., 2017; Ouyang et al., 2022), removing the value model and deriving advantages from group-relative rewards. At each training step, given a prompt $\mathbf{x}$ sampled from the dataset $\mathcal{D}$, GRPO samples G responses $\mathcal{G}^{\otimes G} = \{\mathbf{y}_1, \ldots, \mathbf{y}_G\}$ from $\pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{x})$. Let $\mathcal{A}(\mathbf{y}_i)$ denote the answer corresponding to response $\mathbf{y}_i$. Each response receives a scalar reward r_i, forming $\mathbf{r} = (r_1, \ldots, r_G)$, and its advantage is obtained by group-wise normalization:

$$A_i = \frac{r_i - \mu_{\mathbf{r}}}{\sigma_{\mathbf{r}}}, \qquad \mu_{\mathbf{r}} = \frac{1}{G}\sum_{j=1}^{G} r_j, \qquad \sigma_{\mathbf{r}} = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_j - \mu_{\mathbf{r}})^2}. \tag{1}$$

Throughout, zero advantages are assigned whenever $\sigma_{\mathbf{r}} = 0$. The response-level advantage is shared across all tokens, *i.e.*, $A_{i,t} = A_i$. Defining $\rho_{i,t} = \pi_\theta(y_{i,t} \mid \mathbf{x}, \mathbf{y}_{i,<t}) / \pi_{\theta_{\text{old}}}(y_{i,t} \mid \mathbf{x}, \mathbf{y}_{i,<t})$, GRPO optimizes a clipped objective with KL regularization D_{KL}:

$$\mathcal{J}(\theta) = \mathbb{E}_{\mathbf{x}\sim P_{\mathcal{D}},\, \mathcal{G}^{\otimes G}\sim \pi_{\theta_{\text{old}}}^{\otimes G}(\cdot|\mathbf{x})} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|\mathbf{y}_i|} \sum_{t=1}^{|\mathbf{y}_i|} \min\{\rho_{i,t}A_{i,t}, \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\, A_{i,t}\} - \beta D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}}) \right]. \tag{2}$$

Standard RLVR. In Eq.1, the reward r_i of each response $\mathbf{y}_i$ is computed by a reward function $\mathcal{R}(\mathbf{y}_i \mid \xi)$, where ξ denotes the *reward reference*. In standard RLVR, this reference is given by the ground-truth answer a_{true}. The reward is therefore determined solely by the fixed external reference:

$$\xi \equiv a_{\text{true}}, \qquad r_i^{\text{VR}} = \mathcal{R}(\mathbf{y}_i \mid \xi) = \mathbb{1}[\mathcal{A}(\mathbf{y}_i) = a_{\text{true}}]\,. \tag{3}$$

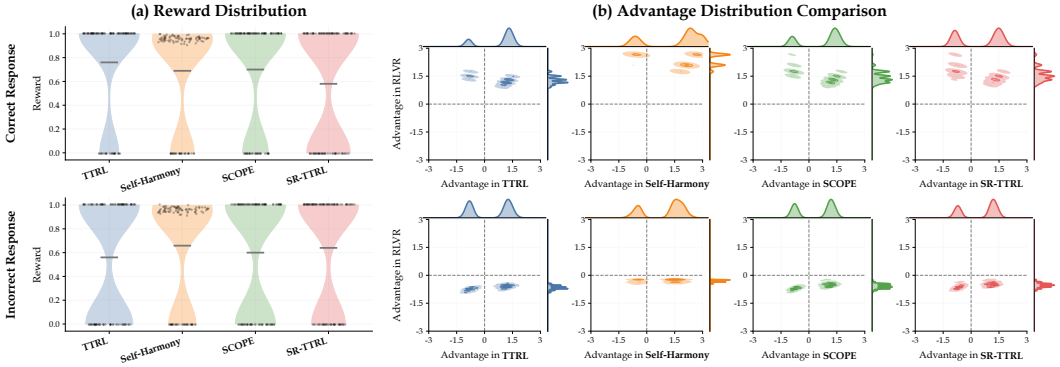


Figure 2: **Self-Reward Uncertainty Demonstration. (a)** Reward distributions for a fixed correct response (top) and a fixed incorrect response (bottom) across group contexts under TTRL (Zuo et al., 2026), Self-Harmony (Wang et al., 2026a), SCOPE (Wang et al., 2026b), and SR-TTRL (Wu et al., 2026). Dots denote individual realizations and gray bars denote means. **(b)** Joint distributions of self-rewarding (x-axis) and RLVR (y-axis) advantages for the same responses. Results here are from `Qwen3-8B-Base` on a `DAPO-14K` prompt; more cases are provided in Appendix A.

Mainstream Pipeline of Self-Rewarding RL. However, a_{true} is unavailable in self-rewarding RL, so ξ must be inferred from the sampled rollout group $\mathcal{G}^{\otimes G} = \{\mathbf{y}_1, \ldots, \mathbf{y}_G\}$. Specifically, a group-dependent function f constructs the reward reference $\xi = f(\mathcal{G}^{\otimes G})$, against which the reward for $\mathbf{y}_i$ is:

$$r_i^{\text{SR}} = \mathcal{R}(\mathbf{y}_i \mid \xi) = \mathcal{R}\big(\mathbf{y}_i \mid f(\mathcal{G}^{\otimes G})\big)\,. \tag{4}$$

A common choice is majority voting (Wang et al., 2023a; Zuo et al., 2026), where ξ is the most frequent answer among $\{\mathcal{A}(\mathbf{y}_1), \ldots, \mathcal{A}(\mathbf{y}_G)\}$, and $\mathcal{R}$ is a binary indicator. Other methods refine f or $\mathcal{R}$ while following the same pipeline. Their details are provided in Appendix B.

3 On the Limitation of Self-Reward Uncertainty

Existing self-rewarding methods differ in their choice of $(f, \mathcal{R})$, but share a common property: the reward assigned to each response depends on the rollout group it was sampled from. This dependence is implicit in Eq.4, where the pseudo-label $f(\mathcal{G}^{\otimes G})$ is constructed from the entire rollout group.

To make this dependence explicit, we fix a response $\mathbf{y}_i$ and decompose its rollout group as

$$\mathcal{G}^{\otimes G} = \{\mathbf{y}_i\} \cup \mathcal{H}_i, \qquad \mathcal{H}_i \triangleq \{\mathbf{y}_j\}_{j \neq i}, \tag{5}$$

where $\mathcal{H}_i$ denotes the **group context** realized alongside $\mathbf{y}_i$. Since all responses are independently sampled from the same policy $\pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{x})$, group contexts follow the same underlying distribution for all focal responses. We therefore use $\mathcal{H}$ to denote a generic random group context. For a fixed response $\mathbf{y}_i$, its reward under $\mathcal{H}$ is

$$r_i(\mathcal{H}) = \mathcal{R}(\mathbf{y}_i \mid f(\{\mathbf{y}_i\} \cup \mathcal{H}))\,, \qquad \mathcal{H} = \{\mathbf{z}_1, \ldots, \mathbf{z}_{G-1}\} \sim Q(\cdot \mid \mathbf{x}) \equiv \pi_{\theta_{\text{old}}}^{\otimes(G-1)}(\cdot \mid \mathbf{x}) \tag{6}$$

Although $r_i(\mathcal{H}_i)$ is deterministic once the realized context $\mathcal{H}_i$ is fixed, $\mathcal{H}$ itself is randomly sampled. Thus, the reward intended to estimate the quality of $\mathbf{y}_i$ depends not only on the response itself but also on its group context, introducing uncertainty into the resulting learning signal.

Empirical Validation. We examine how group-context variation affects the reward and advantage of the same response. Given a model π, we fix a prompt $\mathbf{x}$ and sample one correct focal response $\mathbf{y}^+$ and one incorrect focal response $\mathbf{y}^-$. For each response, we sample $M = 100$ independent group contexts, each containing $G - 1$ newly sampled responses with $G = 16$. These contexts produce M self-reward realizations and group-normalized advantages, forming their empirical distributions. For comparison, we compute the corresponding RLVR advantage under each context.

Figure 2 presents results for four representative self-rewarding methods. As shown in **(a)**, the reward of the same response varies substantially across group contexts. Although the desired signal is positive for a correct response and negative for an incorrect one, both may receive the opposite signal under some contexts. *Thus, restricting reward assignment to a single group-context realization may miss the desired signal, limiting the reliability of self-reward as an estimate of response quality.* In **(b)**, RLVR advantages vary in magnitude but preserve the desired direction: correct responses remain

non-negative, whereas incorrect ones remain non-positive. By contrast, self-rewarding advantages often change sign, causing the same response to be reinforced under one context but suppressed under another. *Therefore, group-context uncertainty in reward assignment can propagate into advantage estimation, making the optimization direction dependent on the sampled context.*

4 GMAE: Group-Marginalized Advantage Estimation

Section 3 shows that a correct response may miss a positive reward signal under a single group context. However, the suitable context is response-dependent and unknown in advance, making it difficult to identify in practice. Therefore, instead of relying on one group context, we aggregate reward realizations across contexts into a distribution-level response representation, improving the chance of retaining desired signals. This idea motivates our **Group-Marginalized Advantage Estimation (GMAE)**. In Section 4.1, we will introduce its core algorithm, and in Section 4.2, we will instantiate it with different self-rewarding rules.

4.1 Core Algorithm

Main Step I: Group-Marginalized Reward Distribution. For a response $\mathbf{y}_i$, instead of using the scalar reward induced by the single realized group context $\mathcal{G}^{\otimes G}/\{\mathbf{y}_i\}$, we consider its reward over the group-context distribution in Eq.6. Let δ_v denote the Dirac measure concentrated at v, we define the *group-marginalized reward distribution* as

$$P_{r_i} = \int \delta_{\mathcal{R}(\mathbf{y}_i | f(\{\mathbf{y}_i\} \cup \mathcal{H}))} \, \mathrm{d}Q(\mathcal{H} \mid \mathbf{x}) = \int \delta_{\mathcal{R}(\mathbf{y}_i | f(\{\mathbf{y}_i, \mathbf{z}_1, \ldots, \mathbf{z}_{G-1}\}))} \prod_{j=1}^{G-1} \mathrm{d}\pi_{\theta_{\text{old}}}(\mathbf{z}_j \mid \mathbf{x}), \tag{7}$$

However, directly computing Eq.7 is intractable because it requires marginalizing over the $(G-1)$-fold product distribution. Naive Monte Carlo estimation can approximate this marginalization, but requires repeatedly sampling fresh group contexts for each response, incurring substantial rollout cost. Therefore, we reuse sampled responses through a shared candidate pool (Wang et al., 2026c).

Specifically, besides the G main responses $\mathcal{G}^{\otimes G} = \{\mathbf{y}_1, \ldots, \mathbf{y}_G\}$ used for policy optimization, we sample G' auxiliary responses to form $\mathcal{C} = \{\mathbf{y}_1, \ldots, \mathbf{y}_{G+G'}\}$. For each main response $\mathbf{y}_i$ $(1 \le i \le G)$, choosing $G-1$ companions from $\mathcal{C} \setminus \{\mathbf{y}_i\}$ yields $\binom{G+G'-1}{G-1}$ possible group contexts. We uniformly sample K distinct contexts $\{\mathcal{S}_i^k\}_{k=1}^K$ up to τ, and construct the *empirical distribution*[1]

$$\widehat{P}_{r_i} = \frac{1}{K} \sum_{k=1}^{K} \delta_{\mathcal{R}(\mathbf{y}_i | f(\{\mathbf{y}_i\} \cup \mathcal{S}_i^k))}, \qquad K = \min\left\{\tau, \binom{G+G'-1}{G-1}\right\}. \tag{8}$$

Now we measure the error from replacing the ideal P_{r_i} in Eq.7 with the empirical $\widehat{P}_{r_i}$ in Eq.8 with the total variation distance $d_{\text{TV}}(\cdot, \cdot)$. For a finite reward-value space $\mathcal{V}_i = \text{supp}(P_{r_i})$, with the confidence level and G fixed, the error scales as

$$d_{\text{TV}}\left(\widehat{P}_{r_i}, P_{r_i}\right) = \mathcal{O}\left(\sqrt{\frac{|\mathcal{V}_i|}{K}} + \sqrt{\frac{|\mathcal{V}_i|}{G+G'}}\right). \tag{9}$$

It decreases with K and G' but increases with $|\mathcal{V}_i|$. The exact bound and proof are in Appendix C.1.

Main Step II: Distribution-based Advantage Estimation. After marginalizing over group contexts, each $\widehat{P}_{r_i}$ defines a response-level reward profile that is no longer tied to any particular group-context realization. Accordingly, GMAE treats these distributions as response-level reward representations and independently draws reward realizations from them for advantage estimation.

Specifically, we construct the product distribution $\bigotimes_{j=1}^{G} \widehat{P}_{r_j}$ over joint reward realizations $\mathbf{v} = (v_1, \ldots, v_G)$, where each $v_j \sim \widehat{P}_{r_j}$ and $\mathbf{v} \in \widehat{\mathcal{V}} = \prod_{j=1}^{G} \widehat{\mathcal{V}}_j$. For each $\mathbf{v}$, we compute the group-normalized advantage and then take its expectation under the product distribution, yielding

$$A_i^{\text{gmae}} = \mathbb{E}_{\mathbf{v} \sim \bigotimes_{j=1}^{G} \widehat{P}_{r_j}} \left[\frac{v_i - \mu_{\mathbf{v}}}{\sigma_{\mathbf{v}}}\right] = \sum_{\mathbf{v} \in \widehat{\mathcal{V}}} \left\{\prod_{j=1}^{G} \left[\frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\left[\mathcal{R}\left(\mathbf{y}_j \mid f\left(\{\mathbf{y}_j\} \cup \mathcal{S}_j^k\right)\right) = v_j\right]\right]\right\} \frac{v_i - \mu_{\mathbf{v}}}{\sigma_{\mathbf{v}}}. \tag{10}$$

[1]To avoid combinatorial explosion in enumeration, we set $\tau = 10^4$ by increasing the number of evaluated contexts tenfold from 10 and observing negligible changes in the empirical reward distribution beyond 10^4.

Attached Mechanism: Advantage Scale Calibration. Although group-marginalized advantages preserve richer reward information, *marginalization can shrink their scale and weaken policy updates*. Since $\sum_{i=1}^{G} A_i^{\text{gmae}} = 0$, their variance is $\text{Var}(\{A_i^{\text{gmae}}\}_{i=1}^{G}) = \frac{1}{G}\sum_{i=1}^{G}(A_i^{\text{gmae}})^2$, which directly reflects the advantage scale and effective update strength. By Jensen's inequality (Jensen, 1906),

$$\text{Var}\big(\{A_i^{\text{gmae}}\}_{i=1}^{G}\big) = \frac{1}{G}\sum_{i=1}^{G}\left(\mathbb{E}_{\mathbf{v}}\left[\frac{v_i-\mu_{\mathbf{v}}}{\sigma_{\mathbf{v}}}\right]\right)^2 \le \mathbb{E}_{\mathbf{v}}\left[\frac{1}{G}\sum_{i=1}^{G}\left(\frac{v_i-\mu_{\mathbf{v}}}{\sigma_{\mathbf{v}}}\right)^2\right] \le 1. \tag{11}$$

The right-hand side is the expected variance of conventional group-wise advantages without marginalization and is at most 1 under normalization. Thus, marginalization often reduces the advantage scale, *motivating calibration to restore the non-marginalized update strength.*

Prompt-level calibration could counteract GMAE's reduced advantage scale by matching each prompt to its scale from a single group context. However, scale differences across prompts can reflect prompt-specific reward uncertainty. A response favored in some contexts but disfavored in others has normalized advantages that partially cancel under marginalization. This reduces the prompt's scale and makes its updates more conservative. Calibrating each prompt separately would erase these uncertainty-dependent differences. We formalize this relation in Appendix C.2.

Therefore, we adopt batch-level calibration to restore the overall scale while preserving relative scale differences across prompts. For a batch of B prompts, let $A_{b,i}^{\text{base}}$ denote the non-marginalized advantage. We denote the batch-level scales of $A_{b,i}^{\text{base}}$ and $A_{b,i}^{\text{gmae}}$ by σ_{base} and σ_{gmae}:

$$\sigma_{\text{base}} = \sqrt{\frac{1}{BG}\sum_{b=1}^{B}\sum_{i=1}^{G}\left(A_{b,i}^{\text{base}}\right)^2}, \qquad \sigma_{\text{gmae}} = \sqrt{\frac{1}{BG}\sum_{b=1}^{B}\sum_{i=1}^{G}\left(A_{b,i}^{\text{gmae}}\right)^2}. \tag{12}$$

We use σ_{base} as the reference because it retains the conventional update scale induced by group-wise normalization. We then calibrate all marginalized advantages using a shared factor λ, with a lower bound of 1 to ensure that calibration only compensates for scale contraction:

$$\widetilde{A}_{b,i}^{\text{gmae}} = \lambda A_{b,i}^{\text{gmae}}, \qquad \lambda = \max\left(1, \frac{\sigma_{\text{base}}}{\sigma_{\text{gmae}}}\right). \tag{13}$$

4.2 Instantiating GMAE with Self-Rewarding Rules

As shown in Section 4.1, GMAE is a general strategy agnostic to the reward-reference construction f and reward function $\mathcal{R}$. Different $(f, \mathcal{R})$ configurations define different self-rewarding rules that provide the "fuel", while GMAE serves as the "vehicle" that better leverages these signals for policy optimization. Naturally, concrete choices of f and $\mathcal{R}$ are required to instantiate GMAE.

We first instantiate $\mathcal{R}$ with the binary reward $\mathcal{R}_{\text{bin}}(\mathbf{y}_i \mid \xi) = \mathbb{1}[\mathcal{A}(\mathbf{y}_i) = \xi]$, where ξ denotes the reward reference. We adopt it for three reasons. First, it follows the standard reward design in RLVR. Second, it restricts $\mathcal{V}_i, \widehat{\mathcal{V}}_i \subseteq \{0, 1\}$, and hence $|\widehat{\mathcal{V}}| \le 2^G$, minimizing the reward-value space enumerated in Eq.10. Third, $|\mathcal{V}_i| \le 2$ tightens the approximation bound in Eq.9. With $\mathcal{R}_{\text{bin}}$ fixed, we combine it with three representative f, yielding three corresponding GMAE instantiations:

GMAE$_0$: Standard Voting. Following TTRL (Zuo et al., 2026), GMAE$_0$ constructs the pseudo-label by majority voting (Wang et al., 2023a) over responses sampled from the current policy:

$$\xi_0 = f_0(\mathcal{G}^{\otimes G}) = \arg\max_a \sum_{j=1}^{G} \mathbb{1}[\mathcal{A}(\mathbf{y}_j) = a]\,. \tag{14}$$

GMAE$_{\text{CR}}$: Co-Refinement. Following Co-Reward (Zhang et al., 2026b), GMAE$_{\text{CR}}$ defines the group context $\mathcal{H}$ as G responses sampled from a slowly updated co-teacher and constructs the reward reference by majority voting over this context, where α_t denotes the teacher retention coefficient:

$$\widetilde{\theta}_t = \alpha_t\widetilde{\theta}_{t-1} + (1-\alpha_t)\theta_{\text{old},t}, \quad \widetilde{\mathcal{H}} = \{\widetilde{\mathbf{y}}_j\}_{j=1}^{G} \sim \pi_{\widetilde{\theta}_t}^{\otimes G}(\cdot \mid \mathbf{x}), \quad f_{\text{CR}}\left(\widetilde{\mathcal{H}}\right) = \arg\max_a \sum_{j=1}^{G} \mathbb{1}[\mathcal{A}(\widetilde{\mathbf{y}}_j) = a]\,. \tag{15}$$

GMAE$_{\text{SR}}$: Self-Refinement. Following SR-TTRL (Wu et al., 2026), GMAE$_{\text{SR}}$ uses self-reflective verification instead of voting. For each distinct answer a, it summarizes a representative response and constructs the reward reference through self-reflective comparison:

Table 1: **Main Results Across Models and Domains:** MEAN@16 scores on eight benchmarks across four base models. *Average* is computed over all eight benchmarks. **Bold** and underlined values denote the **best** and second-best results among all self-rewarding methods, respectively.

Method	Mathematics					Science	Knowledge	Coding	*Average*	Mathematics					Science	Knowledge	Coding	*Average*
	MATH500	AMC	AIME24	AIME25	AIME26	GPQA	MMLU-Pro	LiveCode		MATH500	AMC	AIME24	AIME25	AIME26	GPQA	MMLU-Pro	LiveCode	
	`Qwen3-1.7B-Base`									`Qwen3-4B-Base`								
Raw Model	$45.7_{0.22}$	$17.0_{0.09}$	$2.3_{0.07}$	$0.8_{0.18}$	$1.3_{0.19}$	$7.9_{0.23}$	$13.3_{0.29}$	$6.7_{0.05}$	$11.9_{0.07}$	$50.5_{0.06}$	$24.3_{0.18}$	$8.0_{0.20}$	$3.8_{0.11}$	$5.4_{0.05}$	$15.8_{0.27}$	$14.2_{0.17}$	$15.3_{0.06}$	$17.2_{0.09}$
w/ Verifiable Reward	*64.6*$_{0.08}$	*27.1*$_{0.13}$	*7.1*$_{0.20}$	*5.4*$_{0.18}$	*4.2*$_{0.23}$	*22.6*$_{0.24}$	*31.0*$_{0.23}$	*14.2*$_{0.07}$	*22.0*$_{0.17}$	*81.3*$_{0.16}$	*48.0*$_{0.25}$	*19.3*$_{0.03}$	*17.7*$_{0.11}$	*14.8*$_{0.18}$	*33.0*$_{0.17}$	*39.4*$_{0.32}$	*19.1*$_{0.08}$	*34.1*$_{0.13}$
Intuitor*	$52.1_{0.14}$	$19.8_{0.10}$	$1.1_{0.24}$	$1.2_{0.13}$	$0.8_{0.18}$	$13.3_{0.24}$	$15.7_{0.30}$	$10.7_{0.22}$	$14.3_{0.21}$	$67.1_{0.18}$	$32.9_{0.04}$	$9.2_{0.21}$	$2.7_{0.20}$	$0.7_{0.20}$	$30.0_{0.22}$	$41.1_{0.23}$	$15.8_{0.08}$	$24.9_{0.21}$
EM-RL*	$57.5_{0.08}$	$19.2_{0.23}$	$2.6_{0.18}$	$1.7_{0.13}$	$0.9_{0.22}$	$13.2_{0.32}$	$14.8_{0.19}$	$11.6_{0.08}$	$15.2_{0.06}$	$70.9_{0.19}$	$30.5_{0.18}$	$4.4_{0.09}$	$6.3_{0.04}$	$2.6_{0.08}$	$28.3_{0.19}$	$40.8_{0.25}$	$16.3_{0.15}$	$25.0_{0.21}$
TTRL	$61.1_{0.13}$	$24.0_{0.22}$	$3.4_{0.24}$	$2.7_{0.07}$	$1.1_{0.04}$	$20.2_{0.22}$	$23.7_{0.22}$	$11.3_{0.24}$	$18.4_{0.15}$	$74.6_{0.23}$	$37.8_{0.04}$	$8.8_{0.11}$	$5.0_{0.19}$	$5.3_{0.11}$	$32.9_{0.16}$	$35.4_{0.30}$	$16.0_{0.14}$	$27.0_{0.05}$
CoVo	$52.6_{0.08}$	$19.0_{0.16}$	$1.7_{0.11}$	$1.1_{0.22}$	$1.2_{0.06}$	$12.8_{0.24}$	$17.1_{0.25}$	$10.1_{0.21}$	$14.5_{0.12}$	$60.0_{0.19}$	$29.5_{0.11}$	$7.9_{0.07}$	$5.6_{0.04}$	$4.8_{0.03}$	$28.0_{0.20}$	$31.9_{0.14}$	$15.2_{0.10}$	$22.9_{0.21}$
SCOPE	$56.2_{0.07}$	$22.3_{0.10}$	$4.0_{0.22}$	$2.5_{0.07}$	$1.5_{0.05}$	$12.0_{0.18}$	$13.6_{0.19}$	$8.4_{0.22}$	$15.1_{0.21}$	$73.8_{0.24}$	$38.1_{0.05}$	$9.8_{0.22}$	$6.5_{0.21}$	$6.1_{0.10}$	$31.6_{0.31}$	$30.2_{0.14}$	$16.6_{0.22}$	$26.6_{0.03}$
Self-Harmony	$60.1_{0.08}$	$22.0_{0.21}$	$4.0_{0.20}$	$1.5_{0.12}$	$1.5_{0.21}$	$20.3_{0.23}$	**27.4**$_{0.12}$	$11.4_{0.19}$	$18.5_{0.12}$	$73.6_{0.12}$	$33.6_{0.18}$	$8.4_{0.23}$	$5.3_{0.17}$	$5.3_{0.08}$	$34.8_{0.15}$	$41.7_{0.31}$	$16.2_{0.25}$	$27.4_{0.10}$
Co-Reward	$61.2_{0.23}$	$23.6_{0.21}$	$4.0_{0.05}$	$2.6_{0.21}$	$1.1_{0.15}$	$18.9_{0.27}$	$23.2_{0.27}$	$12.3_{0.25}$	$18.4_{0.06}$	$74.4_{0.06}$	$38.0_{0.13}$	$9.4_{0.19}$	$4.8_{0.12}$	$5.7_{0.15}$	$33.7_{0.25}$	$33.8_{0.13}$	$17.2_{0.09}$	$27.1_{0.10}$
RESTRAIN	$58.3_{0.17}$	$22.3_{0.16}$	$3.6_{0.10}$	$2.1_{0.15}$	$1.9_{0.19}$	$16.2_{0.25}$	$19.8_{0.30}$	$11.7_{0.16}$	$17.0_{0.11}$	$71.0_{0.24}$	$37.2_{0.04}$	$8.2_{0.23}$	$5.5_{0.17}$	$4.9_{0.12}$	$31.0_{0.28}$	$35.2_{0.15}$	$16.9_{0.23}$	$26.2_{0.09}$
SR-TTRL	$60.9_{0.08}$	$24.3_{0.23}$	$4.1_{0.12}$	$2.5_{0.11}$	$0.7_{0.21}$	$20.2_{0.25}$	$24.7_{0.19}$	$11.9_{0.24}$	$18.7_{0.18}$	$74.3_{0.16}$	$38.7_{0.13}$	$9.8_{0.09}$	$5.9_{0.23}$	$5.5_{0.21}$	$32.1_{0.19}$	$31.0_{0.18}$	$16.6_{0.09}$	$26.7_{0.19}$
GMAE$_{\theta}$ (Ours)	$63.0_{0.11}$	$25.8_{0.06}$	$5.5_{0.09}$	$4.2_{0.04}$	$2.5_{0.12}$	**22.2**$_{0.18}$	$25.4_{0.22}$	$14.3_{0.07}$	$20.4_{0.10}$	$75.7_{0.05}$	$39.7_{0.12}$	$10.9_{0.08}$	$8.4_{0.06}$	$6.5_{0.10}$	$34.1_{0.15}$	$38.8_{0.21}$	$17.1_{0.09}$	$28.9_{0.07}$
GMAE$_{\text{CR}}$ (Ours)	**63.8**$_{0.06}$	**26.6**$_{0.10}$	$6.4_{0.05}$	**5.0**$_{0.08}$	$3.0_{0.11}$	$22.0_{0.14}$	$25.0_{0.19}$	$14.5_{0.12}$	$20.8_{0.06}$	**76.2**$_{0.09}$	$40.2_{0.05}$	$11.5_{0.11}$	$8.1_{0.07}$	**7.8**$_{0.04}$	**35.3**$_{0.17}$	**43.4**$_{0.20}$	**18.3**$_{0.10}$	**30.1**$_{0.08}$
GMAE$_{\text{SR}}$ (Ours)	$63.6_{0.08}$	$26.4_{0.05}$	**6.9**$_{0.12}$	$4.6_{0.09}$	**3.3**$_{0.06}$	**22.2**$_{0.20}$	$26.3_{0.16}$	**14.8**$_{0.04}$	**21.0**$_{0.11}$	$75.7_{0.12}$	**41.2**$_{0.08}$	**12.0**$_{0.06}$	**9.2**$_{0.10}$	$7.0_{0.05}$	$34.3_{0.22}$	$40.4_{0.18}$	$17.5_{0.07}$	$29.7_{0.09}$
	`Qwen3-8B-Base`									`Llama3.1-8B-Instruct`								
Raw Model	$68.5_{0.19}$	$35.4_{0.11}$	$11.3_{0.20}$	$7.3_{0.07}$	$6.1_{0.19}$	$31.0_{0.17}$	$42.8_{0.14}$	$19.5_{0.06}$	$27.7_{0.12}$	$47.3_{0.14}$	$21.3_{0.06}$	$7.2_{0.15}$	$1.5_{0.13}$	$1.2_{0.25}$	$17.4_{0.13}$	$21.5_{0.29}$	$9.2_{0.14}$	$15.8_{0.20}$
w/ Verifiable Reward	*83.1*$_{0.25}$	*49.4*$_{0.09}$	*20.4*$_{0.16}$	*19.1*$_{0.20}$	*16.3*$_{0.17}$	*42.9*$_{0.15}$	*71.2*$_{0.30}$	*26.0*$_{0.09}$	*40.9*$_{0.11}$	*69.4*$_{0.11}$	*41.2*$_{0.07}$	*16.2*$_{0.24}$	*7.2*$_{0.14}$	*7.0*$_{0.04}$	*25.8*$_{0.28}$	*35.5*$_{0.26}$	*16.5*$_{0.07}$	*27.4*$_{0.16}$
Intuitor*	$71.1_{0.03}$	$37.8_{0.04}$	$11.9_{0.15}$	$9.7_{0.23}$	$6.8_{0.24}$	$35.2_{0.14}$	$60.6_{0.14}$	$19.8_{0.05}$	$31.6_{0.17}$	$52.1_{0.06}$	$27.6_{0.11}$	$7.7_{0.19}$	$2.7_{0.06}$	$1.6_{0.06}$	$19.9_{0.17}$	$27.6_{0.32}$	$10.2_{0.05}$	$18.7_{0.21}$
EM-RL*	$73.4_{0.12}$	$40.4_{0.25}$	$13.1_{0.22}$	$9.8_{0.25}$	$7.9_{0.10}$	$35.8_{0.29}$	$65.3_{0.22}$	$20.5_{0.09}$	$33.3_{0.22}$	$58.9_{0.15}$	$34.2_{0.18}$	$9.0_{0.07}$	$3.5_{0.17}$	$2.2_{0.04}$	$20.3_{0.24}$	$26.7_{0.27}$	$13.1_{0.07}$	$21.0_{0.17}$
TTRL	$77.8_{0.25}$	$43.6_{0.03}$	$13.1_{0.14}$	$10.5_{0.09}$	$6.1_{0.16}$	$39.7_{0.24}$	$64.1_{0.14}$	$20.9_{0.17}$	$34.5_{0.04}$	$63.8_{0.24}$	$36.6_{0.10}$	$11.6_{0.16}$	$5.0_{0.19}$	$4.1_{0.21}$	$21.1_{0.30}$	$31.8_{0.13}$	$14.3_{0.12}$	$23.5_{0.17}$
CoVo	$69.9_{0.11}$	$37.1_{0.13}$	$11.9_{0.09}$	$9.1_{0.15}$	$7.0_{0.06}$	$32.3_{0.15}$	$52.8_{0.16}$	$20.4_{0.04}$	$30.1_{0.05}$	$51.3_{0.15}$	$29.5_{0.25}$	$10.2_{0.14}$	$2.4_{0.17}$	$1.9_{0.05}$	$18.5_{0.22}$	$29.3_{0.18}$	$12.4_{0.16}$	$19.4_{0.13}$
SCOPE	$77.5_{0.23}$	$43.3_{0.07}$	$12.5_{0.20}$	$11.5_{0.16}$	$7.1_{0.22}$	$38.6_{0.30}$	$59.3_{0.24}$	$19.5_{0.10}$	$33.7_{0.15}$	$60.3_{0.15}$	$32.8_{0.13}$	$12.1_{0.04}$	$4.2_{0.08}$	$3.0_{0.09}$	$20.0_{0.22}$	$30.3_{0.21}$	$11.9_{0.22}$	$21.8_{0.17}$
Self-Harmony	$76.3_{0.08}$	$41.1_{0.22}$	$12.7_{0.15}$	$11.5_{0.18}$	$6.5_{0.10}$	$40.7_{0.30}$	**70.5**$_{0.21}$	$21.9_{0.16}$	$35.2_{0.11}$	$61.7_{0.15}$	$33.6_{0.09}$	$8.0_{0.16}$	$3.9_{0.19}$	$1.8_{0.06}$	$21.8_{0.30}$	$28.4_{0.14}$	$13.0_{0.24}$	$21.5_{0.08}$
Co-Reward	$78.0_{0.16}$	$43.5_{0.15}$	$13.4_{0.06}$	$10.4_{0.05}$	$6.9_{0.25}$	$40.6_{0.23}$	$65.3_{0.14}$	$21.1_{0.15}$	$34.9_{0.17}$	$65.1_{0.12}$	$36.8_{0.09}$	$12.1_{0.19}$	$5.3_{0.20}$	$4.7_{0.09}$	$21.6_{0.27}$	$32.4_{0.24}$	$15.3_{0.19}$	$24.2_{0.23}$
RESTRAIN	$77.3_{0.19}$	$44.8_{0.10}$	$13.8_{0.17}$	$9.8_{0.10}$	$8.1_{0.12}$	$39.0_{0.12}$	$65.0_{0.21}$	$20.7_{0.23}$	$34.8_{0.20}$	$64.2_{0.11}$	$30.6_{0.21}$	$10.9_{0.23}$	$4.3_{0.14}$	$4.0_{0.18}$	$20.7_{0.20}$	$30.8_{0.31}$	$13.9_{0.11}$	$22.4_{0.16}$
SR-TTRL	$77.3_{0.16}$	$44.5_{0.18}$	$13.8_{0.18}$	$11.1_{0.11}$	$7.2_{0.17}$	$39.2_{0.31}$	$63.7_{0.31}$	$20.1_{0.14}$	$34.6_{0.06}$	$65.0_{0.06}$	$36.3_{0.13}$	$12.1_{0.15}$	$5.3_{0.25}$	$5.5_{0.15}$	$22.0_{0.30}$	$31.6_{0.29}$	$13.5_{0.06}$	$23.9_{0.12}$
GMAE$_{\theta}$ (Ours)	$79.8_{0.05}$	$46.2_{0.11}$	$16.2_{0.07}$	$12.5_{0.13}$	$9.2_{0.06}$	**41.7**$_{0.19}$	$67.4_{0.23}$	$23.5_{0.08}$	$37.1_{0.10}$	$66.5_{0.12}$	$38.4_{0.06}$	$13.2_{0.09}$	$6.5_{0.04}$	$5.8_{0.11}$	$22.9_{0.16}$	$32.8_{0.21}$	$15.4_{0.05}$	$25.2_{0.09}$
GMAE$_{\text{CR}}$ (Ours)	**81.6**$_{0.09}$	**47.0**$_{0.04}$	**17.4**$_{0.10}$	**13.5**$_{0.06}$	**9.5**$_{0.12}$	$41.3_{0.15}$	$69.2_{0.18}$	**24.3**$_{0.11}$	**38.0**$_{0.07}$	$67.7_{0.06}$	**39.8**$_{0.10}$	**14.0**$_{0.05}$	**6.6**$_{0.08}$	**6.4**$_{0.13}$	**23.9**$_{0.17}$	**34.2**$_{0.20}$	**16.7**$_{0.09}$	**26.2**$_{0.06}$
GMAE$_{\text{SR}}$ (Ours)	$80.4_{0.07}$	$46.4_{0.12}$	$16.5_{0.04}$	$13.1_{0.09}$	$8.6_{0.05}$	$40.7_{0.21}$	$66.6_{0.17}$	$22.6_{0.06}$	$36.9_{0.11}$	**68.0**$_{0.10}$	$39.2_{0.05}$	$13.6_{0.12}$	$6.1_{0.07}$	$6.0_{0.09}$	$23.3_{0.19}$	$33.7_{0.16}$	$16.2_{0.11}$	$25.8_{0.08}$

$$\mathbf{s}_a = \mathrm{Summ}_{\pi_{\theta_{\text{old}}}}\left(\mathbf{x}, \mathrm{Rep}\left\{\mathbf{y}_j \mid \mathcal{A}(\mathbf{y}_j) = a\right\}\right), \qquad a \in \mathcal{U}, \qquad \mathcal{U} = \{\mathcal{A}(\mathbf{y}_j)\}_{j=1}^{G}.$$
$$\xi_{\text{SR}} = f_{\text{SR}}(\mathcal{G}^{\otimes G}) = \mathrm{Reflect}_{\pi_{\theta_{\text{old}}}}\left(\mathbf{x}, \{(a, \mathbf{s}_a)\}_{a \in \mathcal{U}}\right). \tag{16}$$

Since our core contribution lies in the general GMAE strategy, we only briefly introduce the self-rewarding rules used in these three instantiations here. Their detailed **pseudocode** and **computational complexity analyses** are deferred to Appendix D.

5 EXPERIMENTS

5.1 SETUP

Models. We conduct our main experiments on `Qwen3-1.7B-Base`, `Qwen3-4B-Base`, and `Qwen3-8B-Base` (Yang et al., 2025) to evaluate GMAE across model scales. We also include `Llama3.1-8B-Instruct` (Grattafiori et al., 2024) to assess its adaptability across model families.

Datasets. For training, we use the math dataset `DAPO-14K` (Yu et al., 2026). For evaluation, we select eight benchmarks across mathematics, science, knowledge, and coding domains. For mathematics, we use `MATH500` (Hendrycks et al., 2021), `AMC`, `AIME24`, `AIME25`, and `AIME26`. The out-of-domain (OOD) benchmarks include `GPQA` (Rein et al., 2024) for science, `MMLU-Pro` (Wang et al., 2024b) for knowledge, and `LiveCodeBench-v6` (Jain et al., 2025) for coding.

Baselines. We primarily compare GMAE with *ensemble-based* methods, including TTRL (Zuo et al., 2026), CoVo (Zhang et al., 2026a), SCOPE (Wang et al., 2026b), Self-Harmony (Wang et al., 2026a), RESTRAIN (YU et al., 2026), Co-Reward (Zhang et al., 2026b), and SR-TTRL (Wu et al., 2026). Detailed descriptions are provided in Appendix B. We also include two *probability-based* methods: Intuitor (Zhao et al., 2026b), which uses self-certainty as its reward, and EM-RL (Agarwal et al., 2026), which uses trajectory entropy as its reward. We further include an RLVR reference to measure the remaining gap to verifiable supervision. We faithfully reproduce all baselines using official implementations when available; otherwise, we implement them based on the original papers.

Implementation. We adopt GRPO (Shao et al., 2024) as the RL backbone and train each base model for one epoch with batch size 64. We use a learning rate of 1×10^{-6} with 5% warmup followed by cosine decay (Loshchilov & Hutter, 2017). For each prompt, we sample $G = 16$ main and $G' = 16$ auxiliary responses with temperature $T = 1.0$ and no top-p or top-k truncation, forming a candidate pool of 32 responses. All 32 responses are generated fully in parallel during rollout. We implement all experiments with `verl` on NVIDIA A100 80GB GPUs.

Evaluation. For each evaluation dataset, we report MEAN@16, defined as the average PASS@1 accuracy over 16 sampled responses per problem. The subscript reports the standard deviation of per-problem PASS@1 outcomes aggregated over the evaluation set. We use temperature $T = 0.6$, top-$p = 0.95$, and top-$k = 20$ for sampling. During training, we evaluate checkpoints on `DAPO-500` (Yu

et al., 2026) every 20 steps and at training end, selecting the checkpoint with the highest validation MEAN@16 for final benchmarking.

Fair Cost Control. Training cost is reflected in two aspects: **GPU memory** and **training duration**. For GPU memory, all methods use $G = 16$ responses for policy updates. For a fair comparison, the baselines also generate 32 responses per prompt, with the additional responses used only for reward-reference construction. This enlarged rollout scheme has been shown to improve performance (Zuo et al., 2026). For training duration, it varies with the computation required by each reward construction and is discussed separately in Section 5.3.

5.2 MAIN RESULTS

Overall Comparisons: Robust Improvement and Generalization. In Table 1, GMAE outperforms existing self-rewarding baselines across 30 of 32 model-benchmark combinations. It achieves improvements of 2.0-2.8 points over the strongest baselines, corresponding to 8.0%-12.3% relative gains across different models. Notably, GMAE approaches the performance of RLVR with verifiable rewards. For example, the performance gap on `Qwen3-1.7B-Base` is reduced to only 1.0 point. Moreover, although trained exclusively on mathematics, GMAE maintains strong performance on OOD benchmarks, demonstrating its broad generalization ability.

Controlled Comparisons: GMAE as a Strong General Strategy. To isolate the contribution of GMAE from the underlying self-rewarding rules, we compare each GMAE instantiation with its base version: GMAE_0 with TTRL, GMAE_{CR} with Co-Reward, and GMAE_{SR} with SR-TTRL. Across all comparisons, GMAE consistently improves the corresponding methods, yielding average relative gains of 8.2%, 10.3%, and 9.5%, respectively. Meanwhile, the relative ranking among different self-rewarding rules is largely preserved after applying GMAE, indicating that GMAE improves reward utilization while retaining their individual strengths. Notably, even the weakest GMAE instantiation consistently outperforms the strongest non-GMAE baseline across all evaluated models. These results demonstrate that the improvement comes from the GMAE strategy for leveraging self-reward signals, rather than from any specific self-rewarding rule alone.

5.3 DETAILED TRAINING BEHAVIOR

Training Dynamic: Stability Analysis. Stable training dynamics are essential for sustained self-improvement. Figure 3 tracks performance on the `DAPO-500` validation set throughout training. Several baselines improve initially but then degrade sharply, exhibiting a "rise-then-collapse" pattern. Notably, both probability-based methods show this behavior, suggesting that probability-based supervision can be unreliable (Zhang et al., 2025). In contrast, GMAE improves steadily and converges without collapse, demonstrating substantially more reliable training dynamics.

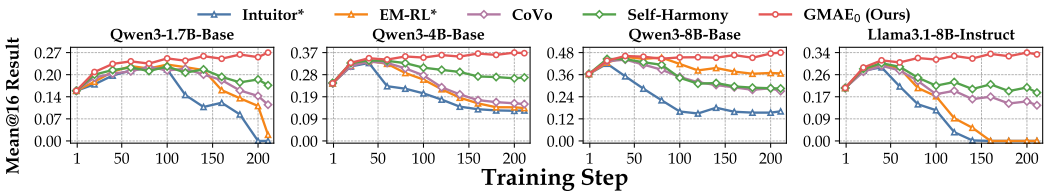


Figure 3: **Training Dynamics** on the `DAPO-500` validation set over training steps.

Training Robustness: Multi-Seed Training Variance. As analyzed in Section 3, randomness in rollout sampling may flip the sign of the estimated advantage for the same response, substantially affecting policy updates. We therefore test whether GMAE produces more consistent outcomes across runs. We evaluate each method over five runs on `Qwen3-8B-Base`, keeping the hardware, initialization, training data, batch order, and hyperparameters fixed while varying only the rollout-sampling seed. Figure 4 reports the standard deviation of MEAN@16 across runs, averaged over eight benchmarks. GMAE_0 achieves the lowest variance at 0.48, with lower variance in every controlled comparison: TTRL → GMAE_0 (0.69 → 0.48), Co-Reward →

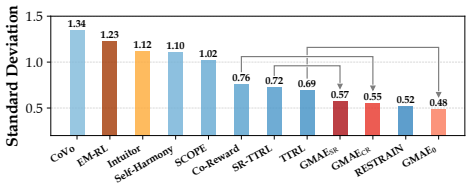


Figure 4: **Multi-Seed Training Variance.** Results show the average MEAN@16 standard deviation across eight benchmarks for `Qwen3-8B-Base` over five training seeds.

GMAE_{CR} (0.76 → 0.55), and SR-TTRL → GMAE_{SR} (0.72 → 0.57). These results show that beyond improving absolute performance, GMAE also makes training more robust.

Training Duration: Cost Analysis. With equal rollout sampling budgets, we compare end-to-end training time under the same hardware and settings, normalized to RLVR. Figure 5 shows that all three GMAE instantiations remain within the low-to-moderate cost range of existing baselines and add little overhead over their corresponding base versions. This limited overhead arises because *marginalization reuses cached responses and requires only lightweight scalar computation*. For example, GMAE_0 introduces only $G(K-1)$ lightweight voting operations per prompt and at most 2^G scalar normalizations over TTRL, resulting in nearly identical training time. Meanwhile, the more elaborate GMAE instantiations require additional computation from their underlying self-rewarding rules but achieve correspondingly stronger performance, yielding a favorable cost-performance trade-off.

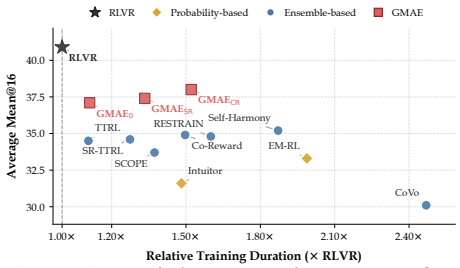


Figure 5: **Training Duration vs. Performance.** Each point represents one method. Duration is normalized to RLVR, and performance is the average MEAN@16 across eight benchmarks on `Qwen3-8B-Base`.

6 Extended Analysis

6.1 Why Does GMAE Work? A Correct-Response Frequency Argument

The key motivation for GMAE is that a correct response may receive a positive reward only in certain group contexts, which a single realization may miss. By retaining reward realizations across contexts, GMAE preserves even sparse positive evidence. Extending this intuition to model performance, we initially expected GMAE to prevent rare correct responses from disappearing while keeping common ones frequent, as they are consistently rewarded across contexts. We test this explanation by tracking correct-response frequencies throughout training.

We compare illustrative count distributions for GMAE_0 and its non-marginalized version TTRL on `Qwen3-8B-Base` at steps 50 and 200, representing early and near-final stages of the 211-step training. As shown in Figure 6, TTRL has more prompts with no correct responses, whereas GMAE_0 has more prompts with a small or moderate number of correct responses. The two distributions have almost the same mass at 16 correct responses. This pattern suggests that GMAE helps keep rare correct responses from disappearing while maintaining success on prompts that are already solved consistently.

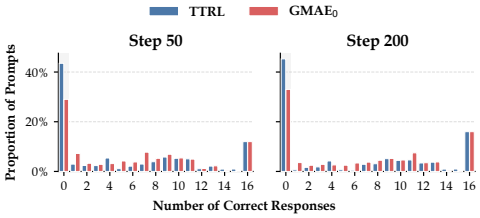


Figure 6: **Distribution of Correct Responses per Prompt.** Illustrative proportions of training prompts by the number of correct responses among $G = 16$ sampled responses under GMAE_0 and TTRL at steps 50 and 200.

6.2 Ablation Study

Auxiliary Response Size G'. Eq.9 establishes that the upper bound on the estimation error of $\hat{P}_{r_i}$ decreases with larger G'. We validate this result empirically by varying G' under otherwise identical configurations and performing multiple runs. When $G' = 0$, the GMAE instantiation reduces to its base version. As shown in Figure 7, increasing G' consistently improves performance, but the marginal benefit drops markedly beyond $G' = 16$. This suggests that our default choice of $G' = 16$ provides a strong balance between performance and rollout cost.

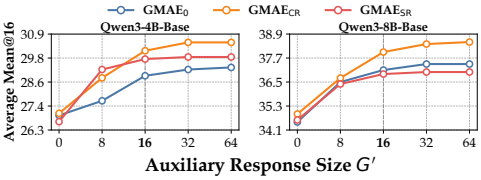


Figure 7: **Auxiliary Response Size Ablation.** Average MEAN@16 across all eight benchmarks with different auxiliary response sizes.

Advantage Scale Calibration Mechanism. Section 4 theoretically motivates advantage scale calibration and explains why calibration should be applied at the batch rather than prompt level. Now we empirically validate this design. Table 2 shows that batch-level calibration performs best on both models, whereas prompt-level calibration performs worst, even underperforming no calibration.

Table 2: **Calibration Ablation on GMAE$_0$.** MEAN@16 results for `Qwen3-8B-Base`.

	MATH500	AMC	AIME24	AIME25	AIME26	GPQA	MMLU-Pro	LiveCode	*Average*
w/o Calibration	78.4	45.2	16.0	10.3	**9.7**	39.2	63.6	22.4	35.6
w/ Prompt-level Calibration	78.0	44.7	15.5	10.8	8.6	39.4	63.8	21.3	35.3
w/ Batch-level Calibration (Ours)	**79.8**	**46.2**	**16.2**	**12.5**	9.2	**41.7**	**67.4**	**23.5**	**37.1**

Figure 8 further shows how the calibration factor λ (Eq.13) evolves during training. Its value remains between 1.15 and 1.35, confirming that GMAE consistently requires moderate scale compensation. The factor gradually decreases as training proceeds, indicating that the gap between the non-marginalized and marginalized advantage scales narrows over time.

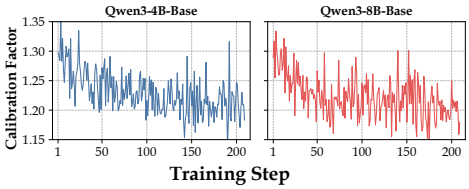


Figure 8: **Calibration Factor λ Dynamics.**

6.3 SCALABILITY STUDY

We evaluate GMAE's scalability along two dimensions: **RL backbones** and **training datasets**. For the former, we replace the default GRPO backbone with GSPO (Zheng et al., 2025) and REINFORCE++ (Hu et al., 2025), keeping all other settings fixed. For the latter, we train all methods on `Open-RS` (Dang & Ngo, 2026) and `MATH-8K` (Hendrycks et al., 2021), in addition to `DAPO-14K`. As shown in Table 3, GMAE consistently retains its advantages over representative strong baselines from Table 1, demonstrating scalability across the two training settings.

Table 3: **Scalability Across Training Settings.** Average MEAN@16 across all eight benchmarks for `Qwen3-8B-Base`. Full benchmark-wise results are in Appendix E.

Method	RL Backbones		Training Datasets	
	GSPO	REINFORCE++	`Open-RS`	`MATH-8K`
TTRL	35.8	32.8	34.0	33.2
Self-Harmony	36.1	33.6	34.9	33.0
Co-Reward	37.3	33.9	35.3	34.3
SR-TTRL	36.8	33.6	35.2	34.3
GMAE$_0$ (Ours)	38.1	36.4	36.2	35.5
GMAE$_{CR}$ (Ours)	**39.3**	**37.5**	**37.3**	36.2
GMAE$_{SR}$ (Ours)	38.7	37.1	36.6	**36.5**

7 RELATED WORK

Self-Evolving LLMs. Self-evolving LLMs improve through self-generated experience (Tao et al., 2024; Gao et al., 2025). They evolve data through self-generation or instruction evolution (Wang et al., 2023b; Xu et al., 2024) and failure-targeted augmentation (Lee et al., 2024b); reasoning through iterative refinement (Madaan et al., 2023; Shinn et al., 2023), latent-rationale bootstrapping (Zelikman et al., 2022; 2024), and structure discovery (Zhou et al., 2024); and policies or tasks through reinforced self-training, self-play (Gulcehre et al., 2023; Chen et al., 2024), and proposer–solver co-evolution (Zhao et al., 2026a; Huang et al., 2026). Yet many require demonstrations, labels, environmental feedback, or verifiers. We instead study *zero-label* evolution from prompts alone, without ground-truth answers or external reward models.

Self-Rewarding RL. Self-rewarding RL uses self-generated supervision, extending RLAIF (Bai et al., 2022; Lee et al., 2024a) to closed self-evaluation loops (Yuan et al., 2024; Wang et al., 2024a; Wu et al., 2025). For reasoning, probability-based methods score individual rollouts using confidence or self-certainty (Zhao et al., 2026b; Li et al., 2025), while entropy-based objectives reward concentrated token distributions (Agarwal et al., 2026). Ensemble-based methods derive pseudo-supervision from multiple rollouts, beginning with answer-frequency consensus (Zuo et al., 2026) and extending to trajectory consistency (Zhang et al., 2026a), weighted or subgroup voting (Wang et al., 2026b), transformed-view agreement (Wang et al., 2026a), complementary or teacher-guided consensus (Zhang et al., 2026b), and self-reflection (Wu et al., 2026). In contrast, GMAE supports different self-reward rules and addresses their shared dependence on sampled rollout groups.

8 CONCLUSION

In this paper, we introduced GMAE, a general advantage-estimation strategy for self-rewarding RL. By marginalizing rewards over possible group contexts, GMAE provides more reliable optimization signals while remaining compatible with different self-rewarding rules. Experiments across multiple models and benchmarks demonstrate consistent improvements and stable training, supporting GMAE as an effective approach to zero-label self-evolution.

AI USE STATEMENT

In this work, we used generative AI tools for writing polishing and the retrieval of some related work. We have verified the correctness of the AI-generated content and take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. The unreasonable effectiveness of entropy minimization in llm reasoning. *Advances in Neural Information Processing Systems*, 38:107150–107180, 2026.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

Quy-Anh Dang and Chris Ngo. Reinforcement learning for reasoning in small LLMs: What works and what doesn't. In *Logical and Symbolic Reasoning in Language Models @ AAAI 2026*, 2026.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Bingxiang He, Yuxin Zuo, Zeyuan Liu, Shangziqi Zhao, Zixuan Fu, Junlin Yang, Cheng Qian, Kaiyan Zhang, Yuchen Fan, Ganqu Cui, et al. How far can unsupervised rlvr scale llm training? In *International Conference on Learning Representations*, volume 2026, pp. 14823–14865, 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. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization. *arXiv preprint arXiv:2501.03262*, 2025.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. In *International Conference on Learning Representations*, volume 2026, pp. 130770–130790, 2026.

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Johan Ludwig William Valdemar Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30(1):175–193, 1906.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. In *Forty-first International Conference on Machine Learning*, 2024a.

Nicholas Lee, Thanakul Wattanawong, Sehoon Kim, Karttikeya Mangalam, Sheng Shen, Gopala Anumanchipalli, Michael Mahoney, Kurt Keutzer, and Amir Gholami. Llm2llm: Boosting llms with novel iterative data enhancement. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6498–6526, 2024b.

Pengyi Li, Matvey Skripkin, Alexander Zubrey, Andrey Kuznetsov, and Ivan Oseledets. Confidence is all you need: Few-shot rl fine-tuning of language models. *arXiv preprint arXiv:2506.06395*, 2025.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Shuvendu Roy, Hossein Hajimirsadeghi, Mengyao Zhai, and Golnoosh Samei. You need reasoning to learn reasoning: The limitations of label-free rl in weak base models. *arXiv preprint arXiv:2511.04902*, 2025.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models. *arXiv preprint arXiv:2404.14387*, 2024.

Ru Wang, Wei Huang, Qi Cao, Yusuke Iwasawa, Yutaka Matsuo, and Jiaxian Guo. Self-harmony: Learning to harmonize self-supervision and self-play in test-time reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 83112–83144, 2026a.

Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. Self-taught evaluators. *arXiv preprint arXiv:2408.02666*, 2024a.

Weiqin Wang, Yile Wang, Kehao Chen, and Hui Huang. Beyond majority voting: Towards fine-grained and more reliable reward signal for test-time reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37251–37265, 2026b.

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*, 2023a.

Yiming Wang, Zhuosheng Zhang, and Rui Wang. On the overscaling curse of parallel thinking: System efficacy contradicts sample efficiency. *arXiv preprint arXiv:2601.21619*, 2026c.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 13484–13508, 2023b.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024b.

Sitong Wu, Haoru Tan, Xichen Zhang, Bin Xia, Shaofeng Zhang, Xiaojuan Qi, Bei Yu, and Jiaya Jia. Beyond majority voting: Self-reflective test-time reinforcement learning for LLM reasoning. In *Forty-third International Conference on Machine Learning*, 2026.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason E Weston, and Sainbayar Sukhbaatar. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 11548–11565, 2025.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Learning Representations*, volume 2024, pp. 30745–30766, 2024.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

ZHAONING YU, Zhaolun Su, Leitian Tao, Haozhu Wang, Aashu Singh, Hanchao Yu, Jianyu Wang, Hongyang Gao, Weizhe Yuan, Jason E Weston, Ping Yu, and Jing Xu. RESTRAIN: From spurious votes to signals — self-training RL with self-penalization. In *The Fourteenth International Conference on Learning Representations*, 2026.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57905–57923. PMLR, 21–27 Jul 2024.

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

Eric Zelikman, Georges Raif Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah Goodman. Quiet-STar: Language models can teach themselves to think before speaking. In *First Conference on Language Modeling*, 2024.

Kongcheng Zhang, Qi Yao, Shunyu Liu, Yingjie Wang, Baisheng Lai, Jieping Ye, Mingli Song, and Dacheng Tao. Consistent paths lead to truth: Self-rewarding reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:59849–59887, 2026a.

Yanzhi Zhang, Zhaoxi Zhang, Haoxiang Guan, Yilin Cheng, Yitong Duan, Chen Wang, Yue Wang, Shuxin Zheng, and Jiyan He. No free lunch: Rethinking internal feedback for llm reasoning. *arXiv preprint arXiv:2506.17219*, 2025.

Zizhuo Zhang, Jianing Zhu, Xinmu Ge, Zihua Zhao, Xuan Li, Xiao Feng, Jiangchao Yao, Bo Han, et al. Co-rewarding: Stable self-supervised rl for eliciting reasoning in large language models. In *International Conference on Learning Representations*, volume 2026, pp. 82520–82558, 2026b.

Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. *Advances in Neural Information Processing Systems*, 38:105816–105879, 2026a.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. In *International Conference on Learning Representations*, volume 2026, pp. 2548–2581, 2026b.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Denny Zhou, Swaroop Mishra, Huaixiu S Zheng, et al. Self-discover: Large language models self-compose reasoning structures. *Advances in Neural Information Processing Systems*, 37:126032–126058, 2024.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Xinwei Long, Ermo Hua, Biqing Qi, et al. Ttrl: Test-time reinforcement learning. *Advances in Neural Information Processing Systems*, 38:131459–131483, 2026.

A Additional Demonstrations of Self-Reward Uncertainty

We provide four additional cases following the setup of Figure 2. For each `DAPO-14K` prompt, we sample one correct and one incorrect focal response from `Qwen3-8B-Base`. Each focal response is evaluated under $M = 100$ independent group contexts, each containing $G - 1$ newly sampled responses with $G = 16$. We record its reward and group-normalized advantage under the four self-rewarding methods, together with its RLVR advantage under the same context.

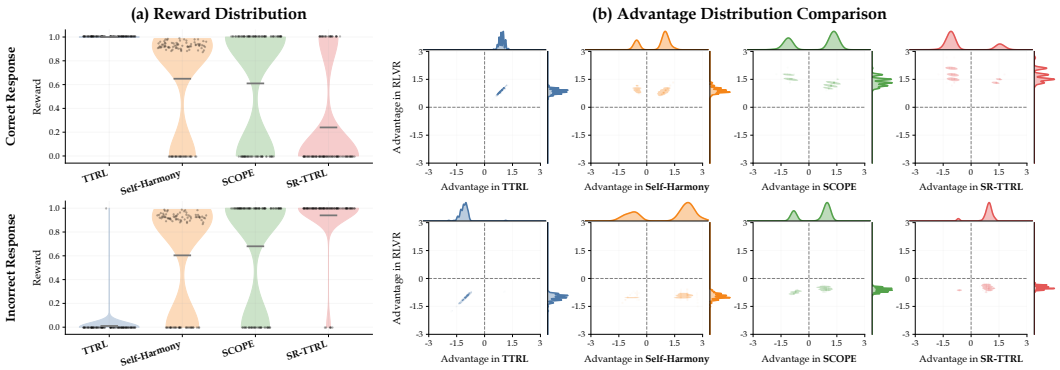


Figure 9: **Self-Reward Uncertainty (Additional Case 2).**

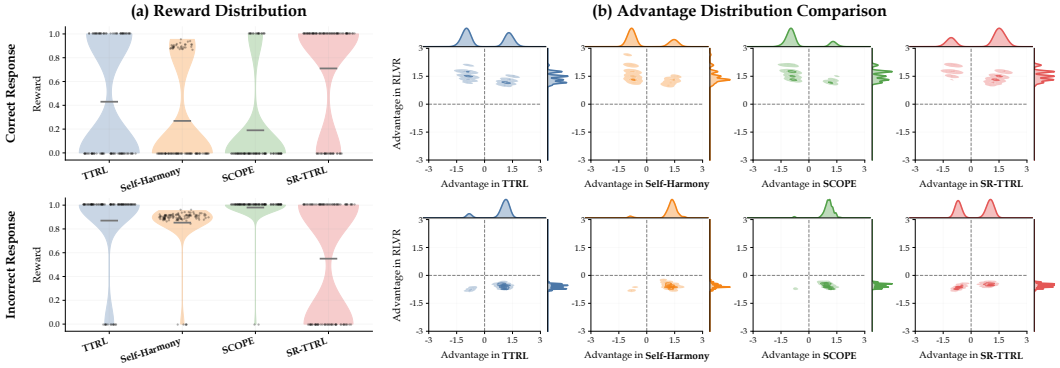


Figure 10: **Self-Reward Uncertainty (Additional Case 3).**

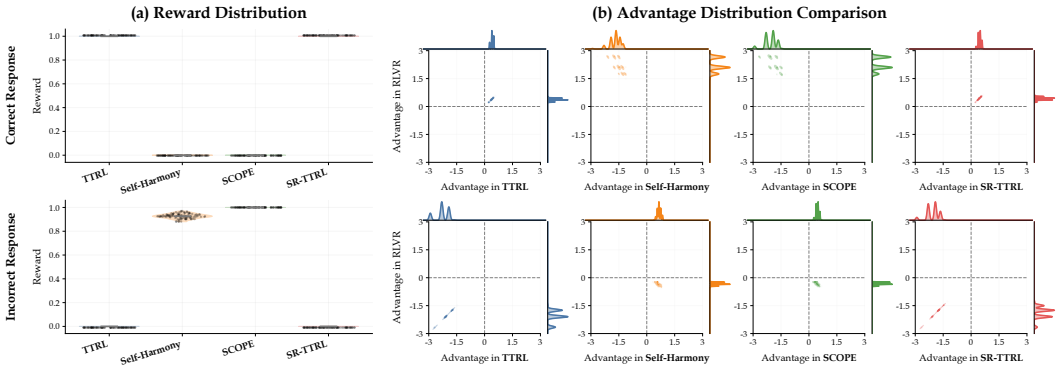


Figure 11: **Self-Reward Uncertainty (Additional Case 4).**

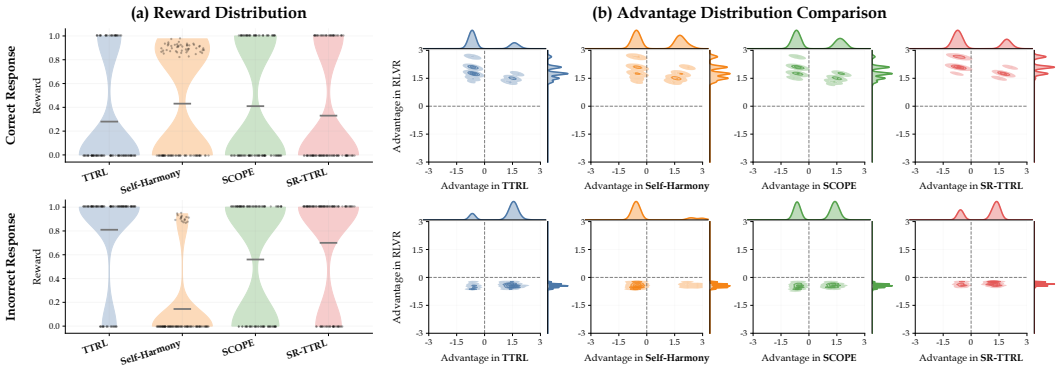


Figure 12: **Self-Reward Uncertainty (Additional Case 5).**

B Self-Rewarding Rules of Baselines

We describe the self-rewarding rules used by the baselines. Following Section 2, let $\mathcal{G}^{\otimes G} = \{\mathbf{y}_1, \ldots, \mathbf{y}_G\}$ be the sampled response group, where $\mathbf{y}_i$ denotes a complete response and $\mathcal{A}(\mathbf{y}_i)$ extracts its final answer. Let

$$\mathcal{U} = \{\mathcal{A}(\mathbf{y}_i)\}_{i=1}^{G}, \qquad n(a) = \sum_{i=1}^{G} \mathbb{1}[\mathcal{A}(\mathbf{y}_i) = a] \tag{17}$$

denote the set of distinct answers and the frequency of answer a, respectively. In general, each method constructs a reward reference $z = f(\mathcal{G}^{\otimes G})$ and assigns $r_i = \mathcal{R}(\mathbf{y}_i \mid z)$. Depending on the method, z may be a single pseudo-label, a collection of response-specific pseudo-labels, or structured group information.

B.1 TTRL

TTRL (Zuo et al., 2026) uses majority voting over the extracted answers. The most frequent answer is selected as the reward reference:

$$\boxed{z_{\mathrm{TTRL}} = f_{\mathrm{TTRL}}(\mathcal{G}^{\otimes G}) = \arg\max_{a \in \mathcal{U}} \sum_{j=1}^{G} \mathbb{1}[\mathcal{A}(\mathbf{y}_j) = a]} \tag{18}$$

Each response receives a binary reward according to whether its final answer matches this reference:

$$\boxed{\mathcal{R}_{\mathrm{TTRL}}(\mathbf{y}_i \mid z_{\mathrm{TTRL}}) = \mathbb{1}[\mathcal{A}(\mathbf{y}_i) = z_{\mathrm{TTRL}}]} \tag{19}$$

B.2 CoVo

CoVo (Zhang et al., 2026a) derives rewards from the internal reasoning trajectory of each response. Specifically, response $\mathbf{y}_i$ is divided into T_i reasoning steps, and $\mathbf{s}_{i,\ell}$ denotes its intermediate state before step ℓ. For a candidate answer a, CoVo measures its distance from this state by

$$d(\mathbf{s}_{i,\ell}, a) = -\frac{1}{|a|}\sum_{q=1}^{|a|}\log \pi_{\theta_{\text{old}}}(a_q \mid \mathbf{s}_{i,\ell}, a_{<q})\,. \tag{20}$$

Let $a_i = \mathcal{A}(\mathbf{y}_i)$. The consistency and volatility of $\mathbf{y}_i$ are defined as

$$\text{Con}(\mathbf{y}_i) = \frac{1}{T_i}\sum_{\ell=0}^{T_i-1}\mathbb{1}\left[a_i = \arg\min_{a\in\mathcal{U}} d(\mathbf{s}_{i,\ell}, a)\right], \tag{21}$$

$$\text{Vol}(\mathbf{y}_i) = \frac{1}{T_i}\max\left(\left\{\ell : a_i \neq \arg\min_{a\in\mathcal{U}} d(\mathbf{s}_{i,\ell}, a)\right\}\cup\{0\}\right). \tag{22}$$

Responses sharing the same extracted answer form

$$\mathcal{I}(a) = \{j : \mathcal{A}(\mathbf{y}_j) = a\}\,. \tag{23}$$

CoVo maps each response to

$$\mathbf{v}_i = \text{Con}(\mathbf{y}_i)\begin{bmatrix}\cos(\text{Vol}(\mathbf{y}_i))\\ \sin(\text{Vol}(\mathbf{y}_i))\end{bmatrix} \tag{24}$$

and aggregates these vectors within each answer group. Accordingly, its group-derived reference is

$$\boxed{z_{\text{CoVo}} = f_{\text{CoVo}}(\mathcal{G}^{\otimes G}) = \left\{\mathcal{I}(a), \{\mathbf{v}_j\}_{j\in\mathcal{I}(a)}\right\}_{a\in\mathcal{U}}} \tag{25}$$

The intrinsic reward of $\mathbf{y}_i$ is the normalized magnitude of the aggregated vector for its answer group:

$$r_i^{\text{int}} = \frac{1}{|\mathcal{I}(a_i)|}\left\|\sum_{j\in\mathcal{I}(a_i)}\mathbf{v}_j\right\|_2. \tag{26}$$

CoVo additionally assigns a curiosity reward r_i^{cur} based on the likelihood of transitions between successive reasoning states, together with a KL penalty that prevents a few extremely unlikely tokens from dominating the reward. The final reward is

$$\boxed{\mathcal{R}_{\text{CoVo}}(\mathbf{y}_i \mid z_{\text{CoVo}}) = r_i^{\text{int}} + r_i^{\text{cur}}} \tag{27}$$

The token-level computation of r_i^{cur} follows the original implementation of CoVo.

B.3 SCOPE

SCOPE (Wang et al., 2026b) combines step-wise confidence weighting with subgroup-specific pseudo-labels. Suppose response $\mathbf{y}_i$ contains L_i reasoning steps, with step ℓ containing $N_{i,\ell}$ tokens. Let $P_{i,\ell,t}^{(1)}, \ldots, P_{i,\ell,t}^{(K_c)}$ denote the top-K_c next-token probabilities at token position t. The response-level confidence is

$$c_i = \frac{1}{L_i}\sum_{\ell=1}^{L_i}\frac{1}{N_{i,\ell}}\sum_{t=1}^{N_{i,\ell}}\left(-\frac{1}{K_c}\sum_{q=1}^{K_c}\log P_{i,\ell,t}^{(q)}\right). \tag{28}$$

SCOPE partitions the G responses into equal-sized subgroups. The subgroup size is dynamically selected through the Pareto criterion in the original method, which balances the agreement between responses and their subgroup labels against the diversity of labels across subgroups. Let $\mathcal{I}_1, \ldots, \mathcal{I}_L$ denote the resulting partition and let $\ell(i)$ satisfy $i \in \mathcal{I}_{\ell(i)}$.

For each subgroup ℓ, SCOPE independently draws a bootstrap sample $\mathcal{B}_\ell$ from the complete rollout group and applies confidence-weighted voting:

$$z_\ell = \arg\max_{a\in\mathcal{U}}\sum_{j\in\mathcal{B}_\ell} c_j\,\mathbb{1}[\mathcal{A}(\mathbf{y}_j) = a]\,. \tag{29}$$

Therefore, SCOPE constructs one reward reference for each response according to its subgroup:

$$\boxed{z_{\mathrm{SCOPE}} = f_{\mathrm{SCOPE}}(\mathcal{G}^{\otimes G}) = \big(z_{\ell(1)}, \ldots, z_{\ell(G)}\big)} \tag{30}$$

Writing $z_{\mathrm{SCOPE},i} = z_{\ell(i)}$, its reward is

$$\boxed{\mathcal{R}_{\mathrm{SCOPE}}(\mathbf{y}_i \mid z_{\mathrm{SCOPE}}) = \mathbb{1}[\mathcal{A}(\mathbf{y}_i) = z_{\mathrm{SCOPE},i}]} \tag{31}$$

The candidate subgroup sizes, bootstrap configuration, and Pareto selection procedure follow the original SCOPE implementation.

B.4 Self-Harmony

Self-Harmony (Wang et al., 2026a) constructs two views of the same prompt. Besides the original prompt $\mathbf{x}$, the model acts as a Reframer to generate a semantically equivalent prompt $\mathbf{x}'$. It then samples an additional response group

$$\mathcal{G}'^{\otimes G'} = \{\mathbf{y}'_1, \ldots, \mathbf{y}'_{G'}\} \sim \pi_{\theta_{\mathrm{old}}}^{\otimes G'}(\cdot \mid \mathbf{x}'). \tag{32}$$

For each candidate answer a, its empirical frequencies in the original and reframed views are

$$p(a) = \frac{1}{G}\sum_{i=1}^{G} \mathbb{1}[\mathcal{A}(\mathbf{y}_i) = a], \qquad p'(a) = \frac{1}{G'}\sum_{j=1}^{G'} \mathbb{1}\big[\mathcal{A}(\mathbf{y}'_j) = a\big]. \tag{33}$$

Self-Harmony selects the answer with the largest harmonic mean of these two frequencies:

$$\boxed{z_{\mathrm{SH}} = f_{\mathrm{SH}}\Big(\mathcal{G}^{\otimes G}, \mathcal{G}'^{\otimes G'}\Big) = \arg\max_a \frac{2p(a)p'(a)}{p(a) + p'(a)}} \tag{34}$$

The score is defined as zero when $p(a) + p'(a) = 0$. The original responses are rewarded by agreement with this shared reference:

$$\boxed{\mathcal{R}_{\mathrm{SH}}(\mathbf{y}_i \mid z_{\mathrm{SH}}) = \mathbb{1}[\mathcal{A}(\mathbf{y}_i) = z_{\mathrm{SH}}]} \tag{35}$$

Self-Harmony also trains the reframing branch using format and diversity rewards. Their prompt templates and implementation details follow the original paper.

B.5 RESTRAIN

RESTRAIN (YU et al., 2026) retains all distinct answers instead of selecting only the majority answer. Write $\mathcal{U} = \{a_1, \ldots, a_M\}$ and define

$$c_m = \sum_{i=1}^{G} \mathbb{1}[\mathcal{A}(\mathbf{y}_i) = a_m], \qquad p_m = \frac{c_m}{G}. \tag{36}$$

Given the Gaussian shaping function

$$g(p) = \exp\left(-\frac{(p-k)^2}{2\sigma^2}\right), \qquad k \in [0,1], \quad \sigma > 0, \tag{37}$$

RESTRAIN assigns each candidate answer the normalized weight

$$w_m = \frac{g(p_m)}{\sum_{\ell=1}^{M} g(p_\ell)}. \tag{38}$$

Its reward reference is therefore a weighted collection of candidate answers:

$$\boxed{z_{\mathrm{RES}} = f_{\mathrm{RES}}(\mathcal{G}^{\otimes G}) = \{(a_m, w_m)\}_{m=1}^{M}} \tag{39}$$

Unlike methods using a single pseudo-label, RESTRAIN evaluates each response against every candidate answer:

$$\boxed{\mathcal{R}_{\mathrm{RES}}(\mathbf{y}_i \mid z_{\mathrm{RES}}) = \{\mathbb{1}[\mathcal{A}(\mathbf{y}_i) = a_m]\}_{m=1}^{M}} \tag{40}$$

Let $r_{i,m} = \mathbb{1}[\mathcal{A}(\mathbf{y}_i) = a_m]$ and let $A_{i,m}$ be the group-normalized advantage computed from $\{r_{j,m}\}_{j=1}^{G}$. When the majority count $c_{\max} = \max_m c_m$ falls below a threshold κ, RESTRAIN sets the rewards to zero and subtracts a fixed offset δ from every advantage:

$$\widetilde{r}_{i,m} = \begin{cases} r_{i,m}, & c_{\max} \geq \kappa, \\ 0, & c_{\max} < \kappa, \end{cases} \qquad \widetilde{A}_{i,m} = \begin{cases} A_{i,m}, & c_{\max} \geq \kappa, \\ A_{i,m} - \delta, & c_{\max} < \kappa. \end{cases} \tag{41}$$

The resulting GRPO objectives are combined using the pseudo-label weights:

$$\mathcal{J}_{\mathrm{RES}}(\theta) = u_{\mathbf{x}} \sum_{m=1}^{M} w_m \mathcal{J}_{\mathrm{GRPO}}\left(\theta; \{\widetilde{A}_{i,m}\}_{i=1}^{G}\right), \tag{42}$$

where $u_{\mathbf{x}}$ is a fixed prompt weight obtained from the self-consistency of a frozen reference model, as defined in the original RESTRAIN implementation.

B.6 Co-Rewarding

Co-Rewarding (Zhang et al., 2026b) constructs its reward reference using a slowly evolving teacher policy. At training step t, the teacher parameters are updated by

$$\widetilde{\theta}_t = \alpha_t \widetilde{\theta}_{t-1} + (1 - \alpha_t)\theta_{\mathrm{old},t}, \tag{43}$$

where $\alpha_t \in [0, 1]$ is the EMA coefficient. The teacher samples

$$\widetilde{\mathcal{G}}^{\otimes \widetilde{G}} = \{\widetilde{\mathbf{y}}_1, \ldots, \widetilde{\mathbf{y}}_{\widetilde{G}}\} \sim \pi_{\widetilde{\theta}_t}^{\otimes \widetilde{G}}(\cdot \mid \mathbf{x}). \tag{44}$$

The majority answer in this teacher-generated group becomes the reward reference:

$$\boxed{z_{\mathrm{CR}} = f_{\mathrm{CR}}\left(\widetilde{\mathcal{G}}^{\otimes \widetilde{G}}\right) = \arg\max_a \sum_{j=1}^{\widetilde{G}} \mathbb{1}[\mathcal{A}(\widetilde{\mathbf{y}}_j) = a]} \tag{45}$$

The student responses are evaluated against this reference:

$$\boxed{\mathcal{R}_{\mathrm{CR}}(\mathbf{y}_i \mid z_{\mathrm{CR}}) = \mathbb{1}[\mathcal{A}(\mathbf{y}_i) = z_{\mathrm{CR}}]} \tag{46}$$

B.7 SR-TTRL

SR-TTRL (Wu et al., 2026) constructs its pseudo-label through three stages: exploration, summarization, and reflection. Given the sampled group $\mathcal{G}^{\otimes G} = \{\mathbf{y}_1, \ldots, \mathbf{y}_G\}$, the exploration stage first collects the distinct extracted answers

$$\mathcal{U} = \{\mathcal{A}(\mathbf{y}_j)\}_{j=1}^{G}. \tag{47}$$

For each $a \in \mathcal{U}$, Rep selects a representative response from those producing answer a. The summarization operator $\mathrm{Summ}_{\pi_{\theta_{\mathrm{old}}}}$ then uses the current prompt and this representative response to produce a concise reasoning summary $\mathbf{s}_a$. Finally, $\mathrm{Reflect}_{\pi_{\theta_{\mathrm{old}}}}$ jointly compares all answer–summary pairs and returns the selected answer as the pseudo-label:

$$\begin{aligned} &\mathbf{s}_a = \mathrm{Summ}_{\pi_{\theta_{\mathrm{old}}}}\left(\mathbf{x}, \mathrm{Rep}\left\{\mathbf{y}_j \mid \mathcal{A}(\mathbf{y}_j) = a\right\}\right), \qquad a \in \mathcal{U}, \qquad \mathcal{U} = \{\mathcal{A}(\mathbf{y}_j)\}_{j=1}^{G}, \\ &\boxed{z_{\mathrm{SR}} = f_{\mathrm{SR}}(\mathcal{G}^{\otimes G}) = \mathrm{Reflect}_{\pi_{\theta_{\mathrm{old}}}}\left(\mathbf{x}, \{(a, \mathbf{s}_a)\}_{a \in \mathcal{U}}\right)}. \end{aligned} \tag{48}$$

Here, Rep denotes the representative-response selection procedure, Summ summarizes the reasoning supporting each candidate answer, and $\mathrm{Reflect}$ compares these summarized reasoning paths to select the final reward reference. The exact summarization and reflection prompts follow the original SR-TTRL implementation.

Each sampled response is then evaluated against this reference using a binary reward:

$$\boxed{\mathcal{R}_{\mathrm{SR}}\left(\mathbf{y}_i \mid z_{\mathrm{SR}}\right) = \mathbb{1}\left[\mathcal{A}(\mathbf{y}_i) = z_{\mathrm{SR}}\right]}. \tag{49}$$

C THEORY OF GMAE

C.1 APPROXIMATION ERROR OF SHARED-POOL ESTIMATION

Here, we formalize the approximation error between the ideal group-marginalized reward distribution P_{r_i} in Eq.7 and its shared-pool estimate $\widehat{P}_{r_i}$ in Eq.8. Let d_{TV} denote total variation distance and assume that $\mathcal{V}_i = \mathrm{supp}(P_{r_i})$ is finite.

Theorem 1 (Approximation Error of Shared-Pool Estimation) *For any $\delta \in (0,1)$, with probability at least $1-\delta$ over the candidate-pool and context sampling,*

$$d_{\mathrm{TV}}\left(\widehat{P}_{r_i}, P_{r_i}\right) \le \sqrt{|\mathcal{V}_i|\log 2 + \log(4/\delta)}\left(\frac{1}{\sqrt{2K}} + \sqrt{\frac{G-1}{G+G'-1}}\right). \tag{50}$$

Proof. Condition on the fixed response $\mathbf{y}_i$ and define

$$N = G + G' - 1, \qquad m = G - 1, \qquad q = \left\lfloor \frac{N}{m} \right\rfloor. \tag{51}$$

Let $\mathbf{z}_1, \ldots, \mathbf{z}_N$ denote the remaining candidate responses and

$$\mathfrak{S}_{N,m} = \{S \subseteq \{1,\ldots,N\} : |S| = m\} \tag{52}$$

denote the set of all valid companion-index subsets. For each $S \in \mathfrak{S}_{N,m}$, define the resulting reward as

$$\rho_i(S) = \mathcal{R}(\mathbf{y}_i \mid f(\{\mathbf{y}_i\} \cup \{\mathbf{z}_j : j \in S\})). \tag{53}$$

Averaging over all contexts in the candidate pool gives the complete-pool distribution

$$\overline{P}^{\mathcal{C}}_{r_i} = \binom{N}{m}^{-1} \sum_{S \in \mathfrak{S}_{N,m}} \delta_{\rho_i(S)}. \tag{54}$$

If $S_1, \ldots, S_K$ are sampled uniformly without replacement from $\mathfrak{S}_{N,m}$, the empirical distribution used by GMAE is

$$\widehat{P}_{r_i} = \frac{1}{K}\sum_{k=1}^{K} \delta_{\rho_i(S_k)}. \tag{55}$$

By the triangle inequality,

$$\begin{aligned} d_{\mathrm{TV}}\left(\widehat{P}_{r_i}, P_{r_i}\right) \le\ & d_{\mathrm{TV}}\left(\widehat{P}_{r_i}, \overline{P}^{\mathcal{C}}_{r_i}\right) \\ & + d_{\mathrm{TV}}\left(\overline{P}^{\mathcal{C}}_{r_i}, P_{r_i}\right). \end{aligned} \tag{56}$$

The first term is the context-subsampling error caused by evaluating only K contexts, while the second is the finite-pool error caused by using N sampled candidate responses.

For distributions supported on the finite space $\mathcal{V}_i$,

$$d_{\mathrm{TV}}(P,Q) = \sup_{\mathcal{A} \subseteq \mathcal{V}_i} |P(\mathcal{A}) - Q(\mathcal{A})|. \tag{57}$$

We first bound the context-subsampling error. For any $\mathcal{A} \subseteq \mathcal{V}_i$, define

$$X_k^{\mathcal{A}} = \mathbb{1}[\rho_i(S_k) \in \mathcal{A}]. \tag{58}$$

Conditioned on the candidate pool, $\widehat{P}_{r_i}(\mathcal{A})$ is the mean of K values sampled without replacement from the finite population $\{\mathbb{1}[\rho_i(S) \in \mathcal{A}] : S \in \mathfrak{S}_{N,m}\}$, whose population mean is $\overline{P}^{\mathcal{C}}_{r_i}(\mathcal{A})$. Hoeffding's inequality for sampling without replacement therefore gives

$$\Pr\left[\left|\widehat{P}_{r_i}(\mathcal{A}) - \overline{P}^{\mathcal{C}}_{r_i}(\mathcal{A})\right| \ge \epsilon_1 \,\middle|\, \mathcal{C}\right] \le 2\exp\left(-2K\epsilon_1^2\right). \tag{59}$$

Applying a union bound over the at most $2^{|\mathcal{V}_i|}$ subsets of $\mathcal{V}_i$ yields

$$\Pr\Big[d_{\mathrm{TV}}\Big(\widehat{P}_{r_i}, \overline{P}^{\mathcal{C}}_{r_i}\Big) \geq \epsilon_1\Big] \leq 2^{|\mathcal{V}_i|+1} \exp\big(-2K\epsilon_1^2\big)\,. \tag{60}$$

The bound is unconditional because its right-hand side does not depend on the realization of $\mathcal{C}$.

We next bound the finite-pool error. For each $\mathcal{A} \subseteq \mathcal{V}_i$, define the symmetric kernel

$$h_{\mathcal{A}}(\mathbf{z}_1, \ldots, \mathbf{z}_m) = \mathbb{1}[\mathcal{R}(\mathbf{y}_i \mid f(\{\mathbf{y}_i, \mathbf{z}_1, \ldots, \mathbf{z}_m\})) \in \mathcal{A}]\,. \tag{61}$$

Its population mean is

$$p_{\mathcal{A}} = \mathbb{E}[h_{\mathcal{A}}(\mathbf{z}_1, \ldots, \mathbf{z}_m)] = P_{r_i}(\mathcal{A}). \tag{62}$$

Moreover,

$$\overline{P}^{\mathcal{C}}_{r_i}(\mathcal{A}) = \binom{N}{m}^{-1} \sum_{S \in \mathfrak{S}_{N,m}} h_{\mathcal{A}}(\mathbf{z}_j : j \in S), \tag{63}$$

which is a U-statistic of order m.

To derive its concentration, let Π_N be the set of all permutations of $\{1, \ldots, N\}$. For each $\pi \in \Pi_N$, divide its first qm indices into q disjoint blocks of size m and define

$$V_{\pi,\mathcal{A}} = \frac{1}{q} \sum_{\ell=1}^{q} h_{\mathcal{A}}\big(\mathbf{z}_{\pi((\ell-1)m+1)}, \ldots, \mathbf{z}_{\pi(\ell m)}\big)\,. \tag{64}$$

Every m-subset appears equally often among these blocks over all permutations, so

$$\overline{P}^{\mathcal{C}}_{r_i}(\mathcal{A}) = \frac{1}{N!} \sum_{\pi \in \Pi_N} V_{\pi,\mathcal{A}}. \tag{65}$$

For any $\lambda > 0$, Jensen's inequality gives

$$\begin{aligned} &\mathbb{E} \exp\Big[\lambda \Big(\overline{P}^{\mathcal{C}}_{r_i}(\mathcal{A}) - p_{\mathcal{A}}\Big)\Big] \\ &\quad \leq \frac{1}{N!} \sum_{\pi \in \Pi_N} \mathbb{E} \exp[\lambda\,(V_{\pi,\mathcal{A}} - p_{\mathcal{A}})]\,. \end{aligned} \tag{66}$$

For a fixed permutation, the q blocks are disjoint and hence their kernel evaluations are independent variables in $[0, 1]$, each with mean $p_{\mathcal{A}}$. Hoeffding's lemma therefore implies

$$\mathbb{E} \exp[\lambda\,(V_{\pi,\mathcal{A}} - p_{\mathcal{A}})] \leq \exp\left(\frac{\lambda^2}{8q}\right). \tag{67}$$

Combining this with Eq.66 and applying the Chernoff bound gives

$$\Pr\Big[\overline{P}^{\mathcal{C}}_{r_i}(\mathcal{A}) - p_{\mathcal{A}} \geq \epsilon_2\Big] \leq \inf_{\lambda > 0} \exp\left(-\lambda\epsilon_2 + \frac{\lambda^2}{8q}\right) = \exp\big(-2q\epsilon_2^2\big)\,, \tag{68}$$

where the minimum is attained at $\lambda = 4q\epsilon_2$. Applying the same argument to the lower tail yields

$$\Pr\Big[\Big|\overline{P}^{\mathcal{C}}_{r_i}(\mathcal{A}) - P_{r_i}(\mathcal{A})\Big| \geq \epsilon_2\Big] \leq 2\exp\big(-2q\epsilon_2^2\big)\,. \tag{69}$$

A union bound over all subsets of $\mathcal{V}_i$ then gives

$$\Pr\Big[d_{\mathrm{TV}}\Big(\overline{P}^{\mathcal{C}}_{r_i}, P_{r_i}\Big) \geq \epsilon_2\Big] \leq 2^{|\mathcal{V}_i|+1} \exp\big(-2q\epsilon_2^2\big)\,. \tag{70}$$

Let

$$L_\delta = |\mathcal{V}_i| \log 2 + \log(4/\delta) \tag{71}$$

and choose

$$\epsilon_1 = \sqrt{\frac{L_\delta}{2K}}, \qquad \epsilon_2 = \sqrt{\frac{L_\delta}{2q}}. \tag{72}$$

By Eq.60 and Eq.70, each error exceeds its corresponding bound with probability at most $\delta/2$. Thus, with probability at least $1-\delta$,

$$d_{\mathrm{TV}}\left(\widehat{P}_{r_i}, P_{r_i}\right) \leq \sqrt{L_\delta}\left(\frac{1}{\sqrt{2K}} + \frac{1}{\sqrt{2q}}\right). \tag{73}$$

Finally, since $N/m \geq 1$,

$$q = \left\lfloor \frac{N}{m} \right\rfloor \geq \frac{N}{2m}, \tag{74}$$

and therefore

$$\frac{1}{\sqrt{2q}} \leq \sqrt{\frac{m}{N}} = \sqrt{\frac{G-1}{G+G'-1}}. \tag{75}$$

Substituting this inequality into Eq.73 proves Eq.50. □

C.2 Uncertainty-Aware Advantage Scale

For prompt b, let

$$\widehat{P}_b = \bigotimes_{i=1}^{G} \widehat{P}_{r_{b,i}} \tag{76}$$

denote the joint reward distribution. For a reward realization $\mathbf{v} = (v_1, \ldots, v_G) \sim \widehat{P}_b$, define its normalized advantage vector as

$$\mathbf{a}_b(\mathbf{v}) = \begin{cases} \left(\dfrac{v_1 - \mu_{\mathbf{v}}}{\sigma_{\mathbf{v}}}, \ldots, \dfrac{v_G - \mu_{\mathbf{v}}}{\sigma_{\mathbf{v}}}\right), & \sigma_{\mathbf{v}} > 0, \\ \mathbf{0}, & \sigma_{\mathbf{v}} = 0. \end{cases} \tag{77}$$

The corresponding GMAE advantage vector is

$$\mathbf{A}_b^{\mathrm{gmae}} = \mathbb{E}_{\mathbf{v}}\left[\mathbf{a}_b(\mathbf{v})\right]. \tag{78}$$

Since every $\mathbf{a}_b(\mathbf{v})$ is zero-centered, $\sum_{i=1}^{G} A_{b,i}^{\mathrm{gmae}} = 0$, and the prompt-wise GMAE scale satisfies

$$\left(s_b^{\mathrm{gmae}}\right)^2 = \mathrm{Var}\left(\{A_{b,i}^{\mathrm{gmae}}\}_{i=1}^{G}\right) = \frac{1}{G}\left\|\mathbf{A}_b^{\mathrm{gmae}}\right\|_2^2. \tag{79}$$

Let $\mathbf{v}$ and $\mathbf{v}'$ be independent draws from $\widehat{P}_b$. We define the context-induced reward uncertainty as

$$\mathcal{U}_b = \Pr_{\mathbf{v}}(\sigma_{\mathbf{v}} = 0) + \frac{1}{2G}\mathbb{E}_{\mathbf{v},\mathbf{v}'}\left[\left\|\mathbf{a}_b(\mathbf{v}) - \mathbf{a}_b(\mathbf{v}')\right\|_2^2\right]. \tag{80}$$

The first term captures reward realizations with no relative reward information, while the second measures disagreement between relative reward assignments across realizations.

Theorem 2 (Reward Uncertainty and GMAE Scale) *For every prompt b,*

$$\left(s_b^{\mathrm{gmae}}\right)^2 = 1 - \mathcal{U}_b. \tag{81}$$

Consequently, greater context-induced reward uncertainty yields a smaller prompt-wise GMAE scale.

Proof. For any non-degenerate reward realization, group-wise normalization gives

$$\frac{1}{G}\left\|\mathbf{a}_b(\mathbf{v})\right\|_2^2 = \frac{1}{G}\sum_{i=1}^{G}\left(\frac{v_i - \mu_{\mathbf{v}}}{\sigma_{\mathbf{v}}}\right)^2 = 1. \tag{82}$$

When $\sigma_{\mathbf{v}} = 0$, we assign $\mathbf{a}_b(\mathbf{v}) = \mathbf{0}$. Therefore,

$$\frac{1}{G}\mathbb{E}_{\mathbf{v}}\left[\left\|\mathbf{a}_b(\mathbf{v})\right\|_2^2\right] = 1 - \Pr_{\mathbf{v}}(\sigma_{\mathbf{v}} = 0). \tag{83}$$

Because $\mathbf{v}$ and $\mathbf{v}'$ are independent and identically distributed,

$$\begin{aligned}
&\frac{1}{2G}\mathbb{E}_{\mathbf{v},\mathbf{v}'}\left[\|\mathbf{a}_b(\mathbf{v})-\mathbf{a}_b(\mathbf{v}')\|_2^2\right] \\
&\quad=\frac{1}{G}\mathbb{E}_{\mathbf{v}}\left[\|\mathbf{a}_b(\mathbf{v})\|_2^2\right]-\frac{1}{G}\left\|\mathbb{E}_{\mathbf{v}}\left[\mathbf{a}_b(\mathbf{v})\right]\right\|_2^2 \\
&\quad=1-\Pr_{\mathbf{v}}(\sigma_{\mathbf{v}}=0)-\left(s_b^{\mathrm{gmae}}\right)^2.
\end{aligned} \tag{84}$$

Rearranging Eq.84 gives

$$\begin{aligned}
\left(s_b^{\mathrm{gmae}}\right)^2 &= 1-\Bigg[\Pr_{\mathbf{v}}(\sigma_{\mathbf{v}}=0) \\
&\qquad+\frac{1}{2G}\mathbb{E}_{\mathbf{v},\mathbf{v}'}\left[\|\mathbf{a}_b(\mathbf{v})-\mathbf{a}_b(\mathbf{v}')\|_2^2\right]\Bigg] \\
&= 1-\mathcal{U}_b,
\end{aligned} \tag{85}$$

which proves the theorem. □

Theorem 2 explains why advantage signs matter. When a response is favored in some reward realizations but disfavored in others, its normalized advantage changes sign, increasing the disagreement term in Eq.80. Marginalization then causes stronger cancellation and produces a smaller scale, leading to more conservative policy updates.

Prompt-level calibration would independently restore every prompt to its non-marginalized scale and remove these uncertainty-dependent differences. In contrast, batch-level calibration applies one shared factor λ, so

$$\frac{\widetilde{s}_b^{\mathrm{gmae}}}{\widetilde{s}_{b'}^{\mathrm{gmae}}}=\frac{\lambda s_b^{\mathrm{gmae}}}{\lambda s_{b'}^{\mathrm{gmae}}}=\frac{s_b^{\mathrm{gmae}}}{s_{b'}^{\mathrm{gmae}}}, \tag{86}$$

preserving the relative uncertainty-aware scales across prompts.

D Detailed Implementation of GMAE Instantiation

This section details the three GMAE instantiations introduced in Section 4.2. All three use the binary reward

$$\mathcal{R}_{\mathrm{bin}}(\mathbf{y}_i \mid z)=\mathbb{1}[\mathcal{A}(\mathbf{y}_i)=z], \tag{87}$$

where z is the reward reference constructed by the corresponding self-rewarding rule.

Given K reward realizations $\{r_i^k\}_{k=1}^K$ for response $\mathbf{y}_i$, its empirical reward distribution is

$$p_i=\frac{1}{K}\sum_{k=1}^{K} r_i^k, \qquad \widehat{P}_{r_i}=p_i\delta_1+(1-p_i)\delta_0. \tag{88}$$

For $\mathbf{v}=(v_1,\ldots,v_G)\in\{0,1\}^G$, define

$$w_{\mathbf{p}}(\mathbf{v})=\prod_{j=1}^{G} p_j^{v_j}(1-p_j)^{1-v_j}, \qquad h_i(\mathbf{v})=\begin{cases}\dfrac{v_i-\mu_{\mathbf{v}}}{\sigma_{\mathbf{v}}}, & \sigma_{\mathbf{v}}>0,\\ 0, & \sigma_{\mathbf{v}}=0,\end{cases} \tag{89}$$

where

$$\mu_{\mathbf{v}}=\frac{1}{G}\sum_{j=1}^{G} v_j, \qquad \sigma_{\mathbf{v}}=\sqrt{\frac{1}{G}\sum_{j=1}^{G}(v_j-\mu_{\mathbf{v}})^2}. \tag{90}$$

The marginalized advantage can therefore be computed exactly as

$$A_i^{\mathrm{gmae}}=\sum_{\mathbf{v}\in\{0,1\}^G} w_{\mathbf{p}}(\mathbf{v})h_i(\mathbf{v}). \tag{91}$$

The first reward realization is produced by the original group used by the corresponding base method and provides A_i^{base} for calibration. After processing all prompts in a batch, we apply the batch-level calibration in Eq.13.

Each instantiation requires G' auxiliary rollouts beyond its base method. We therefore focus below on the additional operations performed after rollout generation.

Algorithm 1: GMAE$_0$: Standard Voting

Input: Prompt $\mathbf{x}$, policy $\pi_{\theta_{\text{old}}}$, group sizes G, G', and number of contexts K
Output: Advantages $\{A_i^{\text{gmae}}\}_{i=1}^{G}$
Sample $\mathcal{C} = \{\mathbf{y}_1, \ldots, \mathbf{y}_{G+G'}\}$ from $\pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{x})$;
Set $\mathcal{G}^{\otimes G} = \{\mathbf{y}_1, \ldots, \mathbf{y}_G\}$ and $z^1 = f_0(\mathcal{G}^{\otimes G})$;
for $i = 1, \ldots, G$ **do**
  Set $r_i^1 = \mathcal{R}_{\text{bin}}(\mathbf{y}_i \mid z^1)$;
  Sample $K-1$ distinct contexts $\{\mathcal{S}_i^k\}_{k=2}^{K}$ from $\mathcal{C} \setminus \{\mathbf{y}_i\}$, with $|\mathcal{S}_i^k| = G-1$;
  for $k = 2, \ldots, K$ **do**
    Set $\mathcal{B}_i^k = \{\mathbf{y}_i\} \cup \mathcal{S}_i^k$;
    Compute $z_i^k = f_0(\mathcal{B}_i^k)$;
    Set $r_i^k = \mathcal{R}_{\text{bin}}(\mathbf{y}_i \mid z_i^k)$;
  end
  Compute $p_i = K^{-1} \sum_{k=1}^{K} r_i^k$;
end
Compute $\{A_i^{\text{gmae}}\}_{i=1}^{G}$ using Eq.91;
return $\{A_i^{\text{gmae}}\}_{i=1}^{G}$;

D.1 GMAE$_0$: Standard Voting

GMAE$_0$ instantiates GMAE with the majority-voting rule used by TTRL (Zuo et al., 2026). For any response group $\mathcal{B}$, its reward reference is

$$f_0(\mathcal{B}) = \arg\max_a \sum_{\mathbf{y} \in \mathcal{B}} \mathbb{1}[\mathcal{A}(\mathbf{y}) = a]\,. \tag{92}$$

The original rollout group is used as the first realization. For each focal response, the remaining $K-1$ contexts are sampled uniformly without replacement from the shared candidate pool.

Additional Computation. Compared with TTRL, GMAE$_0$ evaluates $K-1$ additional group contexts for each focal response. This adds $G(K-1)$ majority-voting operations and the same number of binary reward evaluations. Since each vote scans G responses, these operations require $\mathcal{O}(G^2(K-1))$ scalar computation. Exact distribution-based advantage estimation further enumerates at most 2^G joint binary reward realizations, requiring $\mathcal{O}(G2^G)$ scalar computation. These operations require neither additional model generation nor backpropagation.

D.2 GMAE$_{\text{CR}}$: Co-Refinement

GMAE$_{\text{CR}}$ follows Co-Rewarding (Zhang et al., 2026b). At training step t, its co-teacher parameters are updated as

$$\widetilde{\theta}_t = \alpha_t \widetilde{\theta}_{t-1} + (1-\alpha_t)\theta_{\text{old},t}, \tag{93}$$

where $\alpha_t \in [0,1]$ is the teacher update coefficient. For a co-teacher response group $\widetilde{\mathcal{B}}$, the reward reference is

$$f_{\text{CR}}(\widetilde{\mathcal{B}}) = \arg\max_a \sum_{\widetilde{\mathbf{y}} \in \widetilde{\mathcal{B}}} \mathbb{1}[\mathcal{A}(\widetilde{\mathbf{y}}) = a]\,. \tag{94}$$

The reward reference depends only on the co-teacher group. Therefore, the same K reward references can be reused for all G main responses.

Additional Computation. The co-teacher update is inherited from Co-Rewarding and therefore introduces no GMAE-specific computation. Compared with Co-Rewarding, GMAE$_{\text{CR}}$ performs $K-1$ additional majority votes over groups of G co-teacher responses. Because each reward reference is shared across all main responses, it also adds $G(K-1)$ binary reward evaluations. Together with exact advantage estimation, the additional scalar complexity is

$$\mathcal{O}\big(G(K-1) + G2^G\big)\,. \tag{95}$$

Algorithm 2: GMAE_{CR}: Co-Refinement

Input: Prompt $\mathbf{x}$, parameters $\theta_{\text{old},t}$ and $\widetilde{\theta}_{t-1}$, coefficient α_t, group sizes G, G', and number of contexts K
Output: Advantages $\{A_i^{\text{gmae}}\}_{i=1}^{G}$
Update $\widetilde{\theta}_t$ using Eq.93;
Sample $\mathcal{G}^{\otimes G} = \{\mathbf{y}_i\}_{i=1}^{G}$ from $\pi_{\theta_{\text{old},t}}(\cdot \mid \mathbf{x})$;
Sample $\widetilde{\mathcal{C}} = \{\widetilde{\mathbf{y}}_j\}_{j=1}^{G+G'}$ from $\pi_{\widetilde{\theta}_t}(\cdot \mid \mathbf{x})$;
Set $\widetilde{\mathcal{B}}^1 = \{\widetilde{\mathbf{y}}_j\}_{j=1}^{G}$;
Sample $K-1$ distinct groups $\{\widetilde{\mathcal{B}}^k\}_{k=2}^{K}$ from $\widetilde{\mathcal{C}}$, with $|\widetilde{\mathcal{B}}^k| = G$;
for $k = 1, \ldots, K$ **do**
  Compute $z_{\text{CR}}^k = f_{\text{CR}}(\widetilde{\mathcal{B}}^k)$;
end
for $i = 1, \ldots, G$ **do**
  for $k = 1, \ldots, K$ **do**
    Set $r_i^k = \mathcal{R}_{\text{bin}}(\mathbf{y}_i \mid z_{\text{CR}}^k)$;
  end
  Compute $p_i = K^{-1} \sum_{k=1}^{K} r_i^k$;
end
Compute $\{A_i^{\text{gmae}}\}_{i=1}^{G}$ using Eq.91;
return $\{A_i^{\text{gmae}}\}_{i=1}^{G}$;

D.3 GMAE_{SR}: Self-Refinement

GMAE_{SR} follows the self-reflective reward construction of SR-TTRL (Wu et al., 2026). Instead of independently performing self-reflection for every group context, it first ranks all candidate answers in the shared pool and then reuses this ranking across contexts.

Given the shared candidate pool $\mathcal{C} = \{\mathbf{y}_1, \ldots, \mathbf{y}_{G+G'}\}$, let

$$\mathcal{U}_{\mathcal{C}} = \{\mathcal{A}(\mathbf{y}_j)\}_{j=1}^{G+G'} \tag{96}$$

denote its set of distinct answers. For each $a \in \mathcal{U}_{\mathcal{C}}$, GMAE_{SR} selects a representative response and summarizes its reasoning:

$$\mathbf{s}_a = \text{Summ}_{\pi_{\theta_{\text{old}}}}\left(\mathbf{x}, \text{Rep}\left\{\mathbf{y}_j \in \mathcal{C} \mid \mathcal{A}(\mathbf{y}_j) = a\right\}\right). \tag{97}$$

It then performs self-reflection once over all answer-summary pairs, producing a ranking $\succ_{\text{SR}}$ over $\mathcal{U}_{\mathcal{C}}$:

$$\succ_{\text{SR}} = \text{Reflect}_{\pi_{\theta_{\text{old}}}}\left(\mathbf{x}, \{(a, \mathbf{s}_a)\}_{a \in \mathcal{U}_{\mathcal{C}}}\right). \tag{98}$$

Here, $a \succ_{\text{SR}} a'$ means that a is ranked above a' by self-reflection. The representative-selection rule, summary prompt, and reflection prompt follow the original SR-TTRL implementation.

For the k-th group context of response $\mathbf{y}_i$, define

$$\mathcal{B}_i^k = \{\mathbf{y}_i\} \cup \mathcal{S}_i^k, \qquad \mathcal{U}_i^k = \left\{\mathcal{A}(\mathbf{y}) \mid \mathbf{y} \in \mathcal{B}_i^k\right\}. \tag{99}$$

Its reward reference is the highest-ranked answer present in this context:

$$z_{\text{SR},i}^k = \max_{\succ_{\text{SR}}} \mathcal{U}_i^k, \qquad r_i^k = \mathcal{R}_{\text{bin}}\left(\mathbf{y}_i \mid z_{\text{SR},i}^k\right). \tag{100}$$

Additional Computation. Let U_{base} and U_{pool} denote the numbers of distinct answers in the original rollout group and the shared candidate pool, respectively. SR-TTRL summarizes U_{base} answer classes and performs one reflection, whereas GMAE_{SR} summarizes U_{pool} classes and also performs only one reflection. Therefore, GMAE_{SR} adds $U_{\text{pool}} - U_{\text{base}} \leq G'$ summary operations and does not increase the number of reflection operations, although the single reflection covers a larger candidate set.

Algorithm 3: GMAE_{SR}: Self-Refinement

Input: Prompt $\mathbf{x}$, policy $\pi_{\theta_{\text{old}}}$, group sizes G, G', and number of contexts K
Output: Advantages $\{A_i^{\text{gmae}}\}_{i=1}^{G}$
Sample $\mathcal{C} = \{\mathbf{y}_1, \ldots, \mathbf{y}_{G+G'}\}$ from $\pi_{\theta_{\text{old}}}(\cdot \mid \mathbf{x})$;
Set $\mathcal{G}^{\otimes G} = \{\mathbf{y}_1, \ldots, \mathbf{y}_G\}$;
Construct the distinct-answer set $\mathcal{U}_{\mathcal{C}}$;
foreach $a \in \mathcal{U}_{\mathcal{C}}$ **do**
  Select $\mathbf{y}_a^{\text{rep}} = \text{Rep}\{\mathbf{y}_j \in \mathcal{C} : \mathcal{A}(\mathbf{y}_j) = a\}$;
  Compute $\mathbf{s}_a = \text{Summ}_{\pi_{\theta_{\text{old}}}}(\mathbf{x}, \mathbf{y}_a^{\text{rep}})$;
end
Compute the global answer ranking $\succ_{\text{SR}}$ using Eq.98;
for $i = 1, \ldots, G$ **do**
  Set $\mathcal{S}_i^1 = \mathcal{G}^{\otimes G} \setminus \{\mathbf{y}_i\}$;
  Sample $K-1$ additional distinct contexts $\{\mathcal{S}_i^k\}_{k=2}^{K}$ from $\mathcal{C} \setminus \{\mathbf{y}_i\}$, with $|\mathcal{S}_i^k| = G-1$;
  for $k = 1, \ldots, K$ **do**
    Set $\mathcal{B}_i^k = \{\mathbf{y}_i\} \cup \mathcal{S}_i^k$;
    Extract the answers $\mathcal{U}_i^k = \{\mathcal{A}(\mathbf{y}) : \mathbf{y} \in \mathcal{B}_i^k\}$;
    Select $z_{\text{SR},i}^k = \max_{\succ_{\text{SR}}} \mathcal{U}_i^k$;
    Set $r_i^k = \mathcal{R}_{\text{bin}}(\mathbf{y}_i \mid z_{\text{SR},i}^k)$;
  end
  Compute $p_i = K^{-1} \sum_{k=1}^{K} r_i^k$;
end
Compute $\{A_i^{\text{gmae}}\}_{i=1}^{G}$ using Eq.91;
return $\{A_i^{\text{gmae}}\}_{i=1}^{G}$;

Once the global ranking is obtained, each additional context requires only extracting its answers and selecting the highest-ranked one. Compared with SR-TTRL, this introduces $G(K-1)$ context-specific ranking lookups and binary reward evaluations. A direct implementation scans at most G responses per context, giving $\mathcal{O}(G^2(K-1))$ scalar computation. Exact distribution-based advantage estimation further requires $\mathcal{O}(G2^G)$ scalar operations. Thus, GMAE_{SR} reuses a single self-reflective ranking across all contexts rather than repeating the expensive reflection procedure K times.

E Full Scalability Analysis Results

Table 4: **Scalability on RL backbone.** Benchmark-wise Mean@16 results on `Qwen3-8B-Base` using GSPO as the RL backbone.

Method	Mathematics					Science	Knowledge	Coding	*Average*
	MATH500	AMC	AIME24	AIME25	AIME26	GPQA	MMLU-Pro	LiveCode	
TTRL	78.8	45.3	14.4	11.3	7.2	40.3	66.6	22.3	35.8
Self-Harmony	78.4	43.2	15.2	11.0	6.7	40.9	**70.8**	22.8	36.1
Co-Reward	80.3	47.7	15.5	12.7	7.9	42.0	68.7	23.2	37.3
SR-TTRL	79.7	47.1	14.7	12.6	8.0	41.4	67.6	22.9	36.8
GMAE_0 (Ours)	81.9	48.7	17.3	14.0	9.0	42.1	67.9	24.2	38.1
GMAE_{CR} (Ours)	**83.3**	**50.0**	**17.6**	**15.0**	**9.7**	**43.8**	69.7	**25.0**	**39.3**
GMAE_{SR} (Ours)	82.4	49.8	17.5	14.5	9.1	43.2	68.5	24.8	38.7

Table 5: **Scalability on RL backbone.** Benchmark-wise MEAN@16 results on `Qwen3-8B-Base` using REINFORCE++ as the RL backbone.

Method	Mathematics					Science	Knowledge	Coding	*Average*
	MATH500	AMC	AIME24	AIME25	AIME26	GPQA	MMLU-Pro	LiveCode	
TTRL	75.9	40.1	10.7	9.7	5.7	36.8	64.3	19.4	32.8
Self-Harmony	75.5	39.6	11.1	10.4	6.3	38.1	**68.3**	19.6	33.6
Co-Reward	76.7	41.5	11.9	11.5	6.6	37.7	64.5	20.4	33.9
SR-TTRL	76.9	40.8	11.7	10.8	6.1	37.4	65.0	20.1	33.6
GMAE$_0$ (Ours)	79.8	45.5	13.8	12.9	9.3	40.2	66.7	22.8	36.4
GMAE$_{CR}$ (Ours)	**80.5**	**47.8**	**14.9**	**14.2**	**10.2**	**41.0**	66.8	**24.3**	**37.5**
GMAE$_{SR}$ (Ours)	80.0	47.1	14.4	13.7	9.2	**41.0**	67.4	23.7	37.1

Table 6: **Scalability on training dataset.** MEAN@16 results on `Qwen3-8B-Base` trained with the `Open-RS` dataset.

Method	Mathematics					Science	Knowledge	Coding	*Average*
	MATH500	AMC	AIME24	AIME25	AIME26	GPQA	MMLU-Pro	LiveCode	
TTRL	76.4	42.7	13.5	10.4	7.6	38.8	62.1	20.3	34.0
Self-Harmony	75.6	41.6	13.7	10.8	7.2	39.3	**69.3**	21.4	34.9
Co-Reward	77.3	43.2	14.1	11.6	7.8	40.4	64.9	22.7	35.3
SR-TTRL	77.0	43.5	13.8	11.4	7.8	40.5	65.7	22.2	35.2
GMAE$_0$ (Ours)	77.8	44.0	14.9	12.3	8.3	41.2	67.9	22.9	36.2
GMAE$_{CR}$ (Ours)	**79.3**	**46.5**	**16.3**	**13.7**	**9.0**	**42.3**	68.5	**23.1**	**37.3**
GMAE$_{SR}$ (Ours)	78.6	46.1	16.2	13.0	8.8	40.3	67.2	22.9	36.6

Table 7: **Scalability on training dataset.** MEAN@16 results on `Qwen3-8B-Base` trained with the `MATH-8K` dataset.

Method	Mathematics					Science	Knowledge	Coding	*Average*
	MATH500	AMC	AIME24	AIME25	AIME26	GPQA	MMLU-Pro	LiveCode	
TTRL	75.5	41.2	12.3	9.5	6.7	37.5	63.2	19.5	33.2
Self-Harmony	76.4	41.5	12.0	9.9	6.5	37.3	61.3	19.4	33.0
Co-Reward	77.0	42.3	13.6	8.9	7.1	38.2	**66.0**	21.2	34.3
SR-TTRL	77.1	42.3	13.6	10.4	7.8	38.6	63.9	21.0	34.3
GMAE$_0$ (Ours)	79.1	43.7	15.0	11.1	8.3	40.1	65.1	21.2	35.5
GMAE$_{CR}$ (Ours)	80.3	**44.0**	**15.7**	12.0	9.0	41.0	64.4	22.9	36.2
GMAE$_{SR}$ (Ours)	**80.8**	43.5	15.4	**12.3**	**9.5**	**41.2**	66.1	**23.2**	**36.5**