Towards Better RL Training Data Utilization via Second-Order Rollout

Anonymous ACL submission

Abstract

Reinforcement Learning (RL) has empowered Large Language Models (LLMs) with strong reasoning capabilities, but vanilla RL mainly focuses on generation capability improvement by training with only first-order rollout (generating multiple responses for a question), and we argue that this approach fails to fully exploit the potential of training data because of the neglect of critique capability training. To tackle this problem, we further introduce the concept of second-order rollout (generating multiple critiques for a response) and propose a unified framework for jointly training generation and critique capabilities. Extensive experiments across various models on math datasets demonstrate that our approach can utilize training data more effectively than vanilla RL and achieve better performance under the same training data. Additionally, we uncover several insightful findings regarding second-order rollout and critique training, such as the importance of label balance in critique training and the noise problem of outcome-based rewards. Our work offers a preliminary exploration of dynamic data augmentation and joint generation-critique training in RL, providing meaningful inspiration for the further advancement of RL training.

1 Introduction

With the widespread application of Reinforcement Learning (RL) in post-training (Guo et al., 2025), Large Language Models (LLMs) have demonstrated remarkable reasoning capabilities, which inspires deeper investigations into RL training for LLMs. However, current RL training predominantly focuses on enhancing generation capability, often neglecting the development of critique capability, which can be a performance bottleneck for further improvement. Saunders et al. (2022) also categorizes model capabilities into *Generation* and *Critique* [1]: (1) *Generation* refers to the ability to produce a correct response to a given question; (2) *Critique* denotes the capacity to judge whether a response is correct and to identify specific errors in a wrong response. Intuitively, these two capabilities are not independent: Wang et al. (2025c) finds that fine-tuning a model using only critique data, even without any explicit generation data, can significantly improve its generation performance. Similarly, better critique performance with only generation training is also witnessed by Wang et al. (2025b). The neglect of critique training (Yu et al., 2025b; Xie et al., 2025) makes us wonder whether current RL training solely on generation capability can fully exploit the potential of training data, and we would like to further explore a joint RL training framework (Ruan et al., 2025; Wang et al., 2025a) for better data utilization.

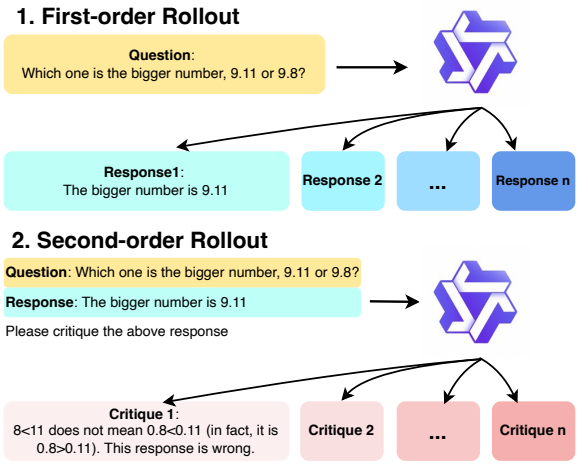

Figure 1: A demonstration of first/second-order rollout. The policy model generates multiple responses for a question in first-order rollout, and generates multiple critiques for a response in second-order rollout.

[1] The original classification delineates three distinct capabilities: Generation, Discrimination, and Critique. For simplicity, we subsume both Discrimination and Critique under the single term Critique.

In §3 we propose **G**eneration and **C**ritique RL (**GC-RL**) that jointly trains two capabilities with only generation training data by introducing the concept of *second-order rollout*. In vanilla RL, a policy model samples multiple responses for a given question during training, which we define as *first-order rollout*. Building on this, we further define *second-order rollout* as the process in which the policy model generates multiple critiques for a <question, response> pair, and these two processes are illustrated in Figure 1. At each training step, we sample questions from the training set for first-order rollout and <question, response> pairs from a data cache for second-order rollout, and these rollout is then combined to update the policy model collectively. Meanwhile, responses generated from first-order rollout are filtered and added to the data cache for subsequent training. It is worth noting that the second-order rollout can proceed naturally based on the results of the first-order rollout, and no additional training data is required, which can be viewed as a "free lunch" from a data perspective.

In §4, extensive experiments are conducted across different models, demonstrating that our GC-RL shows better data utilization and outperforms vanilla RL in both generation and critique capabilities under the same training data. Further we conduct more experiments to explore second-order rollout and critique training in §5 and find: 1. the data filter is crucial for maintaining balanced critique training (§5.1); 2. outcome-based reward is noisy for critique training and denoising can be achieved through multiple samplings (§5.2); 3. static data performs better in critique-only training and dynamic data are more suitable for joint training (§5.3); 4. fine-grained model critique behavior manipulation can be achieved through reward function adjustment. (§5.4).

Our contributions can be summarized as follows:

1. We introduce the concept of *second-order rollout* and propose GC-RL framework, which achieves better RL training data utilization by generation-critique joint training.

2. We conduct extensive experiments to show the effectiveness of our approach and draw some instructive conclusions about critique training.

3. Our work offers a preliminary exploration of dynamic data augmentation in RL training, providing meaningful inspiration for further advancement of RL.

2 Related Work

RL for LLMs The exceptional reasoning capabilities demonstrated by Deepseek-R1(Guo et al., 2025) highlight the significant role of RL(Zhang et al., 2025) with verifiable reward in training LLMs. Beyond rule-based rewards, other forms such as model-based (Xu et al., 2025; Shao et al., 2025) and rubric-based (Gunjal et al., 2025; Huang et al., 2025) rewards can also be leveraged for RL training of LLMs. Another line of research focuses on developing RL algorithms suited for LLM training, including works like PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), DAPO (Yu et al., 2025a), and GSPO (Zheng et al., 2025). There are also works exploring new RL training tasks and objectives: for instance, She et al. (2025) employs RL to train models in reconstructing questions from responses, while Dong et al. (2025) applies RL to enhance next-token prediction. Different from previous works, our work explores better RL training utilization through joint training of generation and critique capabilities.

LLM Critique The ability to provide critique constitutes a crucial component of LLM capabilities. High-quality critiques enable LLMs to perform self-correction (Pan et al., 2024; Yang et al., 2025a,b) more effectively, and can also enhance the reward signals produced by reward models (Yu et al., 2025c; Ankner et al., 2024; Ye et al., 2025) when incorporated into the context. Sun et al. (2024) proposes a framework for evaluating the quality of critiques, while other research efforts have focused on improving critique abilities through Supervised Fine-Tuning (Wang et al., 2025c), Direct Preference Optimization (Yu et al., 2025b) and RL (Xi et al., 2025; Xie et al., 2025; Tang et al., 2025). In contrast to prior work, our approach integrates critique training into the vanilla RL framework.

Data Augmentation Data augmentation (Wang et al., 2024; Chai et al., 2026) is an effective approach to enhance model performance in LLM training. Techniques such as back translation (Sennrich et al., 2016; Kulhánek et al., 2021) and rephrasing (Lu and Lam, 2023) can enrich the dataset without altering semantic meanings. Alternatively, another augmentation strategy involves leveraging LLMs to generate new data in zero-shot (Oh et al., 2023; Ubani et al., 2023) or in-context learning (Dai et al., 2025) settings. Unlike con-

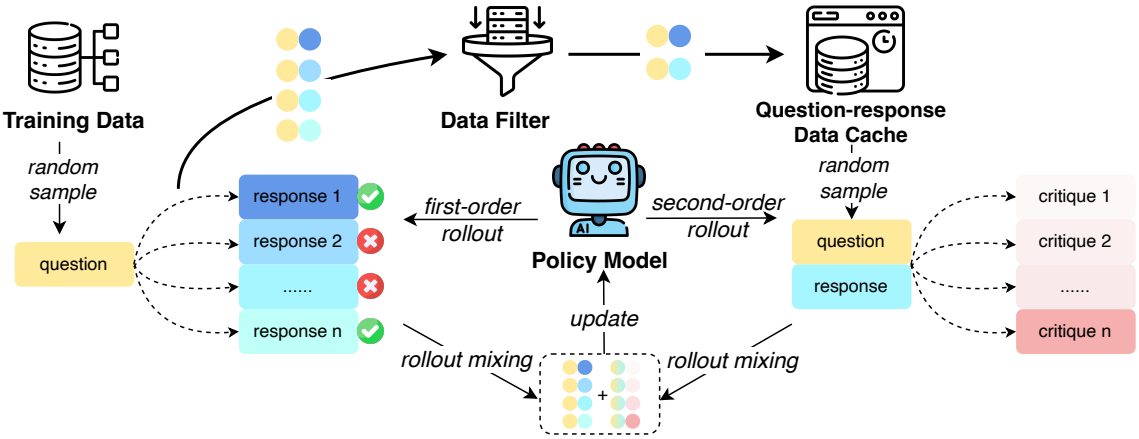

Figure 2: Flowchart of a single training step in GC-RL. First, a batch of questions is sampled from the training data, and multiple responses are generated (first-order rollout). Then, without replacement, a batch of <question, response> pairs is sampled from the *Question-Response Data Cache*, and multiple critiques are generated (second-order rollout). These rollouts are combined and utilized jointly to update the policy model. In addition, the *Question-Response Data Cache* is maintained by processing the first-order rollout through a *Data Filter* and adding the filtered data into the cache.

ventional offline data augmentation methods, our work can be viewed as an online data augmentation process conducted in RL.

3 Methodology

We introduce **G**eneration and **C**ritique **RL** (**GC-RL**), an RL framework for jointly training the generation and critique capabilities, and the overview of our approach is illustrated in the figure 2. At each RL training step, a batch of data is sampled from the initial training set, and the policy model generates multiple responses for each question (in conventional RL training, these responses would be directly used to update the policy model). We go further by feeding these <question, response> pairs into a *data filter*, retaining a subset to be stored in a *question-response data cache*. Then a batch of <question, response> pairs is sampled from the data cache, for which the policy model further generates critiques. The responses and critiques obtained from these two rollout processes are combined, assigned rewards and advantages, and utilized to update the policy model.

RL with Second-order Rollout For each question q_i in the training set D, the policy model samples n responses $r_1, r_2, \ldots, r_n$ during vanilla RL training, a process we refer to as *first-order rollout*. For a specific <question, response> pair $\langle q_i, r_i \rangle$, the policy model further samples n critiques $c_1, c_2, \ldots, c_n$ for subsequent training, which we define as *second-order rollout*. In essence, the first-order rollout aims to enhance the generation capability, enabling model to produce appropriate answers to given questions. The second-order rollout is designed to improve critique capability, empowering it to identify potential issues or errors within a response. It is worth noting that the second-order rollout can proceed naturally based on the results of the first-order rollout, and no additional training data is required, which can be viewed as a "free lunch" from a data perspective.

Critique Data Filter During the RL training process, the <question, response> pairs obtained from the first-order rollout are passed through a *Data Filter*, which controls which pairs to retain. For a question q and its corresponding responses $r_1, r_2, \ldots, r_n$, if all n responses are either entirely correct or entirely incorrect, all these data are discarded. Otherwise, one correct response r_{correct} and one incorrect response r_{wrong} will be randomly selected, and these two resulting pairs $\langle q, r_{correct} \rangle$ and $\langle q, r_{wrong} \rangle$ will then stored in the data cache. Although simple, this data filter plays a crucial role in training stability, and removing this filter would introduce significant issues: (1). *imbalance between critique data and generation data*: for a single question q, the first-order rollout produces n responses and the second-order rollout further generates n^2 critiques, and critique data outnumber generation data by a factor of n; (2). *imbalance within critique data*: we have observed that incorrect responses tend to dominate over correct ones in

the first-order rollout, which can cause a label imbalance problem for further critique training, and further discussion is provided in §5.1.

Mixed Training During the update phase of the policy model, we perform training with a mixture of first-order rollout (responses) and second-order rollout (critiques). For each response r, a rule-based verifier is employed to check its correctness, with the reward function defined as:

$$R(r) = \begin{cases} 1, & r \text{ is correct} \\ 0, & r \text{ is wrong} \end{cases} \quad (1)$$

For each critique c generated based on <question,response> pair $\langle q, r \rangle$, the final judgment regarding the correctness of r is extracted, denoted as $Ext(c) \in \{correct, wrong\}$. Since intermediate steps of a critique are hard to verify (Sun et al., 2024), we only assign an outcome reward based on the final binary judgment, and the corresponding reward function is:

$$R(c) = \begin{cases} 0.7, & Ext(c) = correct \ \& \ R(r)=1 \\ 0.7, & Ext(c) = wrong \ \& \ R(r)=0 \\ 0, & else \end{cases} \quad (2)$$

For responses and critiques in the mixed rollout, the respective rewards are computed with the above reward functions. Then they are mixed into the **same group** to acquire advantages with GRPO (Shao et al., 2024) algorithm for further update of policy model. By mixing data into the same group, we can control the amount of information the model learns from the augmented critique data through scaling reward values of correct critiques, and more discussion is shown in §5.4.

4 Experiments

4.1 Experimental Setup

Models Our experiments are primarily conducted on the Qwen2.5 series (Yang et al., 2024), including Qwen2.5-(1.5B, 3B, 7B)-Base, to demonstrate the effectiveness of our method across models of varying scales. Additional experiments are performed on Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Mistral-7B-Instruct-v0.3 (Rastogi et al., 2025) to further validate the general applicability of our approach to different model architectures. We employ the verl [2] framework for RL training and adopt GRPO algorithm, with training hyper-parameters detailed in the Appendix A.

Dataset The training and evaluation mainly focus on mathematical reasoning tasks, and we leave experiments on more domains to futher work. DAPO-MATH-17k (Yu et al., 2025a) is employed as the primary training dataset: 1k data are randomly selected to construct cold-start data, while the remaining 16k data are utilized for RL training. Several established mathematical reasoning benchmarks are utilized for evaluation, including Math-500(Hendrycks et al., 2021), GSM8k(Cobbe et al., 2021), Minerva(Lewkowycz et al., 2022), AMC23, and OlympiadBench(He et al., 2024).

4.2 RL Training and Experimental Results

Cold Start Applying RL directly to models can lead to several issues: (1). Following the format of CFT (Wang et al., 2025c), we instruct the model to append a correctness judgment of its response at the end of the critique. However, the base models exhibit weak instruction-following capability, often failing to generate critiques that adhere to the required format. (2). Due to their limited reasoning performance, the intermediate reasoning steps within the generated critiques are often of low quality. To address these problems, we first distill 1,885 initial critique data from GPT-5 [3], and the prompt used for critique distillation is provided in the Appendix A. After filtering out data with incorrect formatting or erroneous final judgments, we obtain 1,339 high-quality training examples. Before starting RL, we perform Supervised Fine-Tuning (SFT) on this curated critique dataset to equip the model with a preliminary critique capability.

Baselines In addition to presenting the evaluation results of our **GC-RL** (Generation and Critique RL) approach, we also provide the outcomes of several baseline methods: (1) after cold start without RL training; (2) **G-RL** (Generation RL): vanilla RL training that performs only first-order rollout to enhance generation capability; (3) **C-RL** (Critique RL): for each question, 10 responses are sampled in advance to construct a balanced training set consisting of <question, response> pairs, and only second-order rollout is performed to train critique capability of LLMs. All these RL training are performed under the same training data.

Critique Evaluation In addition to generation capability evaluation, we also try to examine the critique capability after RL training. To construct

[2] https://github.com/volcengine/verl

[3] gpt-5-chat-2025-08-07

Models	Methods	Math-500	GSM8k	Minerva	AMC23	Olympiad	Avg
		Generation Accuracy (%)					
Qwen2.5-7B	w/o RL	55.6	77.9	16.9	35.0	22.8	41.6
	C-RL	65.1	83.7	19.2	47.5	26.1	48.3
	G-RL	75.4	89.7	**24.6**	60.0	33.7	56.7
	GC-RL	**77.6**	**92.0**	**24.6**	**62.5**	**39.8**	**59.3**
Qwen2.5-3B	w/o RL	20.4	29.4	4.0	5.0	7.6	13.3
	C-RL	30.4	50.0	4.4	12.5	11.0	21.7
	G-RL	57.8	79.5	12.9	27.5	24.5	40.4
	GC-RL	**61.8**	**81.4**	**14.9**	**32.5**	**25.2**	**43.2**
Qwen2.5-1.5B	w/o RL	11.0	14.6	1.8	5.0	3.3	7.1
	C-RL	21.3	25.6	3.6	10.0	5.9	13.3
	G-RL	45.1	67.2	8.7	25.0	12.7	31.7
	GC-RL	**47.2**	**69.0**	**10.3**	**27.5**	**15.3**	**33.9**
		Critique Accuracy (%)					
Qwen2.5-7B	C-RL	80.5	82.5	62.3	71.4	72.5	73.8
	GC-RL	**84.6**	**88.3**	**67.1**	**79.4**	**73.8**	**78.6**
Qwen2.5-3B	C-RL	67.4	66.7	**57.1**	64.7	61.2	63.4
	GC-RL	**70.2**	**72.8**	**57.1**	**66.2**	**61.5**	**65.6**
Qwen2.5-1.5B	C-RL	59.5	57.7	53.8	58.8	55.0	57.0
	GC-RL	**61.4**	**60.6**	**57.2**	**61.3**	**57.5**	**59.6**

Table 1: Generation and critique capabilities evaluation results on Qwen-2.5-(1.5B,3B,7B). GC-RL outperforms all other RL training methods in both generation and critique capabilities.

the evaluation dataset, we utilize 5 datasets in generation evaluation as seed data, and sample responses with Qwen2.5-(1.5B,7B,72B)-Instruct, respectively (10 responses for each model). The final answer is required to be enclosed in *boxed{}*, and we filter out responses that dissatisfy this format requirement, as well as questions for which all sampled responses are either entirely correct or entirely incorrect. From the remaining data, we randomly select one correct response and an incorrect one for each question, discarding all other responses, and this process yields critique evaluation datasets in which the correct and incorrect responses are 1:1. Since assessing the accuracy of the intermediate reasoning steps within a critique is particularly challenging, we focus solely on evaluating whether the final judgment of critique on the correctness of the response is accurate, which essentially reduces the task to a binary classification. We also report denoised reward (§5.2) to measure critique capability as supplemental results in Appendix B

Results The evaluation experiments are conducted on both generation and critique capabilities: For generation tasks, model-generated answers are verified by comparing them with reference answers and the final accuracy is reported; For critique tasks, we extract the generated final judgment on the correctness of a response and measure the binary classification accuracy. We show experimental results for Qwen-2.5-(1.5B,3B,7B) in Table 1, and provide more results on Llama-3.1-8B-instruct and Mistral-7B-Instruct-v0.3 in Appendix B, finding that: 1. Even if a model is trained solely with C-RL (without generation training), its generation ability significantly improves—though such enhancement remains far inferior to that achieved through G-RL. 2. After training with GC-RL, the model attains optimal performance in both generation and critique capabilities. On one hand, its generation capability surpasses that of models trained with G-RL; on the other hand, its critique ability exceeds that of models trained with C-RL. This result suggests a certain coupling between critique and generation capabilities, and further demonstrates that joint training yields superior overall performance compared to training for each capability independently.

5 Analysis

First-order rollout in RL training has been thoroughly studied by previous works, so we conduct a more detailed analysis of second-order rollout and critique training in this section. First, we discuss the label imbalance problem in critique training and demonstrate the effectiveness of our data filter both theoretically and experimentally (§5.1). Next, we discuss the reward noise problem in critique train-

		Math-500	GSM8k	Minerva	AMC23	Olympiad	Avg
Genration	Random Sampling	75.0	90.9	22.9	60.0	37.5	57.3
	Random Sampling + Reweight	77.2	91.5	23.2	60.0	38.2	58.0
	Data Filter	**77.6**	**92.0**	**24.6**	**62.5**	**39.8**	**59.3**
Critique	Random Sampling	82.3	84.0	63.3	73.5	71.6	74.9
	Random Sampling + Reweight	83.7	86.7	65.8	77.9	72.8	77.4
	Data Filter	**84.6**	**88.3**	**67.1**	**79.4**	**73.8**	**78.6**

Table 2: A comparison of model performance on Qwen2.5-7B when sampling responses with/without a data filter. Random sampling leads to the worst performance, though adding reward reweighting can alleviate this issue. Utilizing the data filter achieves the best performance.

ing and explore a sampling-based denoising strategy (§5.2). We then compare static and dynamic data in critique training, observing that dynamic data is more suitable for GC-RL, while static data works better for C-RL (§5.3). Finally, by adjusting the reward function, we achieve fine-grained critique behavior manipulation of LLMs after RL training (§5.4).

5.1 Towards Balanced Critique Training

We theoretically analyze why GC-RL without a data filter can lead to training data imbalance and restrict the critique capability of LLMs. To mitigate this problem, we explore employing a reward reweighting strategy and utilizing a data filter, and finally conduct comparative experiments to validate the effectiveness of our data filter.

Alleviating imbalance problem with reward reweighting The ultimate judgement of whether a response is correct or not in a critique is essentially a binary classification task, but the number of erroneous responses significantly outweighs the correct ones in the first-order rollout. Intuitively, this label imbalance problem can impact subsequent critique training, and we also theoretically analyze the effect of data imbalance on the performance of the critique training, with detailed discussions shown in Appendix C. To mitigate this problem, we also explore reweighting and scaling rewards for positive and negative data to balance their contributions, and the weighted reward function is defined as:

$$R_w(c) = \begin{cases} \frac{0.35}{E[R(r)]}, & \textit{Ext(c) = correct \& R(r)=1} \\ \frac{0.35}{1-E[R(r)]}, & \textit{Ext(c) = wrong \& R(r)=0} \\ 0, & else \end{cases} \tag{3}$$

where $E[R(r)]$ is the expected reward for response r during RL training. It can be mathematically proved that the weighted reward is unbiased and does not incentivize the model to judge an uncertain response as correct or wrong, and the detailed proof is shown in Appendix C. Essentially, this weighting strategy amplifies the reward for rare classes, enabling the model to learn more effectively from such data and thereby approximating balanced training.

Empirical comparison of training with/without a data filter Comparison experiments are conducted on Qwen2.5-7B with GC-RL training under three settings: (1) randomly sampling responses without data filter, (2) randomly sampling responses and training with the weighted reward function in Equation 3, and (3) utilizing the data filter in §3. As the experimental results shown in Table 2, random sampling strategy leads to the worst generation and critique capabilities because of label imbalance, and this problem can be alleviated to some extent with a weighted reward function. Although reward weighting can achieve balance in the reward level for imbalanced data, utilizing a data filter can achieve inherently balanced training data and yield the best performance.

5.2 Reward Noise & Denoising in Critique RL

Reward noise problem in critique RL Obtaining precise rewards for critiques is more challenging than for responses. For instance, in mathematical problems, since we have a corresponding answer to each question, the correctness of a response can be easily rule-based verified and a precise reward can then be assigned accordingly. However, for critiques, it is difficult to verify the correctness of each intermediate step, and we can only assign rewards based on whether the final binary classification result is correct. Generating responses is essentially a generation task, and it is rare for a model to produce intermediate errors yet still arrive at the correct final answer. In contrast, generating critiques is essentially a binary classification

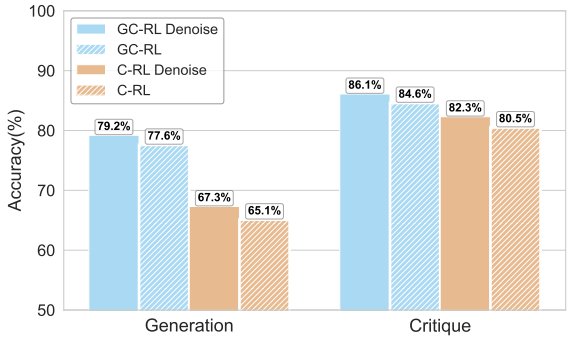

Figure 3: A comparison of model performance of Qwen2.5-7B with/without reward denoising strategy on Math-500. In both GC-RL and C-RL settings, reward denoising can improve model performance on both generation and critique capabilities.

task, and even random guessing can yield correct answers with a 50% probability, leading to many critiques with incorrect intermediate steps but correct final judgement. Ideally, for critiques with correct outcomes, we should differentiate between those with erroneous intermediate steps and those with correct ones, assigning lower and higher rewards, respectively. However, verifying intermediate steps is challenging in practice, so critiques with correct results are often given the same reward, and we refer to such rewards as *noised rewards*. Guo et al. (2025) has demonstrated significant success in Reinforcement Learning with Verifiable Rewards (RLVR), which fundamentally relies on *oracle rewards*. Liu et al. (2025); Shi and Jin (2025); Whitehouse et al. (2025) also attempt RL on classification tasks, showing that even with *noised rewards*, model performance can be improved to some extent, which is also witnessed by our experiments in §4.

Exploring critique reward denoising based on self-correction with multiple samplings In LLM self-correction process (Kamoi et al., 2024; Pan et al., 2024), a model is provided with three components: <question, response, critique>, and then instructed to modify the original response based on the potential issues identified by the critique and generate a refined response. Intuitively, the higher the quality of the critique—i.e., the more accurately it identifies problems in the original response—the greater the likelihood that the refined response will be correct. Thus, we can inversely estimate the quality of a critique based on the quality of its corresponding refined response. Similar to (Tang et al., 2025; Xie et al., 2025; Yu et al., 2025b), for a given critique, we allow the model to perform self-correction and sample n refined responses, with the number of correct ones denoted as k, based on which we then propose the following reward function to estimate critique quality:

$$R_q(c) = \begin{cases} 0.1 * \frac{k}{n}, & \textit{Ext(c) == correct} \\ 0.7 * \frac{k}{n}, & \textit{Ext(c) == wrong} \end{cases} \quad (4)$$

Although it is difficult to directly verify the correctness of intermediate steps in critique, this sampling method can indirectly estimate it to some extent, thereby reducing noise in the reward. We utilize the sum of the outcome reward from Equation 2 and the estimated reward obtained from Equation 4, denoted as $R(c) + R_q(c)$, as the final reward for critique, and compare it with using only the outcome reward $R(c)$. Theoretically, a larger number of samples n leads to better noise reduction and more accurate reward values, but at a higher computational cost. To make the computational overhead controllable, experiments are conducted with $n = 1$, and the experimental results on Math-500 are shown in Figure 3 (with more results shown in Figure 9 in Appendix B). A performance improvement in both generation and critique capabilities can be witnessed under both CG-RL and C-RL settings with our reward denoising strategy.

5.3 Static v.s. Dynamic Data

A comparison of static and dynamic responses during critique RL training. In our approach, dynamic self-generated responses are utilized when generating second-order rollout, and an alternative strategy involves utilizing static responses (Ruan et al., 2025; Xie et al., 2025) for critique RL training, where pre-prepared <question, response> pairs remain fixed throughout the RL process. A performance comparison of static and dynamic training data is conducted under GC-RL and C-RL settings for both generation and critique capabilities on Qwen2.5-7B, and the experimental results on Math-500 are presented in Figure 4 (with more results shown in Figure 7 in Appendix B). We find that training with dynamically self-generated response data leads to higher performance in both generation and critique capabilities under GC-RL setting. However, under C-RL setting, where responses are abandoned and only critiques are utilized to update the policy model, dynamic data suffers from a severe reward hacking problem, and the static data strategy significantly outperforms the dynamic

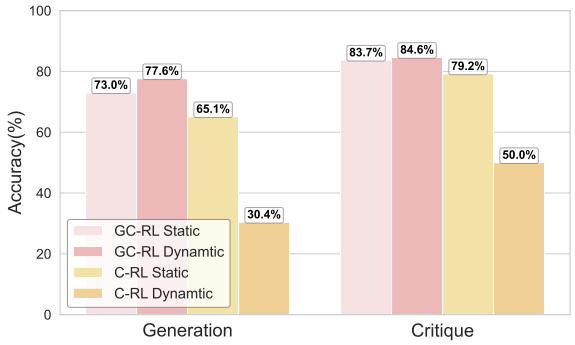

Figure 4: Performance of Qwen2.5-7B on Math-500 under GC-RL and C-RL settings with static and dynamic critique training data. Dynamic data outperforms static data in the GC-RL setting, while the opposite holds for C-RL.

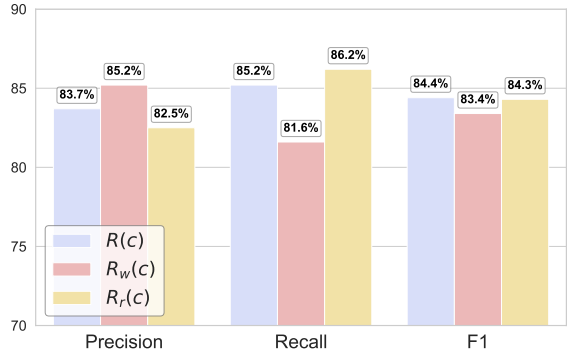

Figure 5: A comparison of critique performance of Qwen2.5-7B with different reward function on Math-500. Compared to baseline $R(c)$, $R_w(c)$ leads to a higher precision while $R_r(c)$ generates a higher recall.

data. With dynamic data, the model seems to identify a shortcut to maximize rewards during RL: it deliberately generates incorrect responses in the generation stage (though producing a correct response is challenging, generating an incorrect one can be quite easy), then it labels all responses as incorrect to get the reward in the critique stage. To summarize, dynamic data is more suitable for GC-RL training, while static data is more appropriate for C-RL training.

5.4 Fine-grained Critique Behavior Manipulation

There are often different requirements for the binary classification performance in different scenarios. For instance, a higher recall rate is demanded in disease screening, while a higher precision rate is required in recommendation systems. In the reward function defined in Equation 2, the same reward is assigned to both correct and incorrect responses as long as the critique identifies them correctly, and we also explore training with more fine-grained reward functions to better control the critique behavior of LLMs. For example, to encourage the model to be more inclined to classify a response as incorrect when it is uncertain, we try the following reward function:

$$R_w(c) = \begin{cases} 0.6, & \textit{Ext(c) = correct \& R(r)=1} \\ 0.8, & \textit{Ext(c) = wrong \& R(r)=0} \\ 0, & else \end{cases} \quad (5)$$

Conversely, to steer the model toward classifying uncertain responses as correct, we assign a large reward value to correct responses and employ the following reward function:

$$R_r(c) = \begin{cases} 0.8, & \textit{Ext(c) = correct \& R(r)=1} \\ 0.6, & \textit{Ext(c) = wrong \& R(r)=0} \\ 0, & else \end{cases} \quad (6)$$

The experimental results on Math-500 of utilizing $R_w(c)$ and $R_r(c)$ are presented in Figure 5, and more results can be found in Figure 8 in Appendix B). Compared with the baseline $R(c)$, we observe that when $R_w(c)$ is applied, the model achieves higher precision but lower recall; whereas with $R_r(c)$, the precision decreases while recall increases. By adjusting the reward function, we can exert finer-grained control over the critique behavior of LLMs.

6 Conclusion

Based on the first-order rollout in vanilla RL (generating multiple responses for a question), we further introduce the concept of second-order rollout (generating multiple critiques for a response) and propose GC-RL, a unified framework to train generation and critique capabilities jointly. Extensive experiments across various models and datasets demonstrate that our approach can more effectively utilize training data compared to vanilla RL, achieving superior performance under the same training data. Additionally, we uncover some insightful findings related to second-order rollout and critique training, such as the importance of label balance in critique training. Our work serves as an initial exploration into dynamic data augmentation and joint training of generation and critique in RL training, offering meaningful insights for further advancements in RL for LLMs.

Limitations

Our work represents a preliminary attempt to integrate dynamic data augmentation with joint training of generation and critique into RL, and still exhibits several limitations that warrant further exploration. For simplicity, we only employ the GRPO algorithm for RL training, and the applicability of our approach to other RL algorithms (such as PPO) remains to be investigated. Theoretically, our method is applicable to any data whose results can be rule-based verified. However, we have only conducted experiments on mathematical tasks, and we would like to leave the verification in other domains for future work. Moreover, experiments are confined to models with fewer than 10B parameters, and extending our approach to larger-scale RL training with multi-domain training data and larger models is considered to be an important direction for future work. Additionally, we have observed that our approach converges more slowly compared to vanilla RL, essentially trading computational resources for improved performance. Our approach also requires that responses can be rule-based verified, making it less straightforward to apply to RL tasks where responses are harder to evaluate (e.g., rubric-based RL training (Gunjal et al., 2025; Huang et al., 2025)). How to perform second-order rollout and assign rewards to critiques in such settings is also worth further exploration.

Ethical Considerations

The data utilized are open for research, and LLMs in the experiments are all publicly available by either parameters or API calls. Therefore, we do not anticipate any ethical concerns in our research.

References

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*.

Yaping Chai, Haoran Xie, and Joe S Qin. 2026. Text data augmentation for large language models: A comprehensive survey of methods, challenges, and opportunities. *Artificial Intelligence Review*, 59(1):35.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Fang Zeng, Wei Liu, and 1 others. 2025. Auggpt: Leveraging chatgpt for text data augmentation. *IEEE Transactions on Big Data*.

Qingxiu Dong, Li Dong, Yao Tang, Tianzhu Ye, Yutao Sun, Zhifang Sui, and Furu Wei. 2025. Reinforcement pre-training. *arXiv preprint arXiv:2506.08007*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean M. Hendryx. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *NeurIPS 2025 Workshop on Efficient Reasoning*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, and 1 others. 2025. Reinforcement learning with rubric anchors. *arXiv preprint arXiv:2508.12790*.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440.

Jonáš Kulhánek, Vojtěch Hudeček, Tomáš Nekvinda, and Ondřej Dušek. 2021. AuGPT: Auxiliary tasks and data augmentation for end-to-end dialogue with pre-trained language models. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 198–210, Online. Association for Computational Linguistics.

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025. Inference-time scaling for generalist reward modeling, 2025. *URL https://arxiv. org/abs/2504.02495*.

Hongyuan Lu and Wai Lam. 2023. Epa: easy prompt augmentation on large language models via multiple sources and multiple targets. *arXiv preprint arXiv:2309.04725*.

Seokjin Oh, Su Ah Lee, and Woohwan Jung. 2023. Data augmentation for neural machine translation using generative language model. *arXiv preprint arXiv:2307.16833*.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506.

Abhinav Rastogi, Albert Q Jiang, Andy Lo, Gabrielle Berrada, Guillaume Lample, Jason Rute, Joep Barmentlo, Karmesh Yadav, Kartik Khandelwal, Khyathi Raghavi Chandu, and 1 others. 2025. Magistral. *arXiv preprint arXiv:2506.10910*.

Chi Ruan, Dongfu Jiang, Yubo Wang, and Wenhu Chen. 2025. Critique-coder: Enhancing coder models by critique reinforcement learning. *arXiv preprint arXiv:2509.22824*.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *CoRR*, abs/2206.05802.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Zhihong Shao, Yuxiang Luo, Chengda Lu, ZZ Ren, Jiewen Hu, Tian Ye, Zhibin Gou, Shirong Ma, and Xiaokang Zhang. 2025. Deepseekmath-v2: Towards self-verifiable mathematical reasoning. *arXiv preprint arXiv:2511.22570*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Shuaijie She, Yu Bao, Yu Lu, Lu Xu, Tao Li, Wenhao Zhu, Shujian Huang, Shanbo Cheng, Lu Lu, and Yuxuan Wang. 2025. Dupo: Enabling reliable llm self-verification via dual preference optimization. *arXiv preprint arXiv:2508.14460*.

Wenlei Shi and Xing Jin. 2025. Heimdall: test-time scaling on the generative verification. *arXiv preprint arXiv:2504.10337*.

Shichao Sun, Junlong Li, Weizhe Yuan, Ruifeng Yuan, Wenjie Li, and Pengfei Liu. 2024. The critique of critique. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9077–9096, Bangkok, Thailand. Association for Computational Linguistics.

Qiaoyu Tang, Hao Xiang, Le Yu, Bowen Yu, Hongyu Lin, Yaojie Lu, Xianpei Han, Le Sun, and Junyang Lin. 2025. Refcritic: Training long chain-of-thought critic models with refinement feedback. *arXiv preprint arXiv:2507.15024*.

Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. Zeroshotdataaug: Generating and augmenting training data with chatgpt. *arXiv preprint arXiv:2304.14334*.

Ke Wang, Jiahui Zhu, Minjie Ren, Zeming Liu, Shiwei Li, Zongye Zhang, Chenkai Zhang, Xiaoyu Wu, Qiqi Zhan, Qingjie Liu, and 1 others. 2024. A survey on data synthesis and augmentation for large language models. *arXiv preprint arXiv:2410.12896*.

Xiaoxuan Wang, Bo Liu, Song Jiang, Jingzhou Liu, Jingyuan Qi, Xia Chen, and Baosheng He. 2025a. From solving to verifying: A unified objective for robust reasoning in llms. *arXiv preprint arXiv:2511.15137*.

Xiyao Wang, Chunyuan Li, Jianwei Yang, Kai Zhang, Bo Liu, Tianyi Xiong, and Furong Huang. 2025b. Llava-critic-r1: Your critic model is secretly a strong policy model. *arXiv preprint arXiv:2509.00676*.

Yubo Wang, Xiang Yue, and Wenhu Chen. 2025c. Critique fine-tuning: Learning to critique is more effective than learning to imitate. In *Second Conference on Language Modeling*.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. J1: Incentivizing thinking in llm-as-a-judge via reinforcement learning. *arXiv preprint arXiv:2505.10320*.

Zhiheng Xi, Jixuan Huang, Xin Guo, Boyang Hong, Dingwen Yang, Xiaoran Fan, Shuo Li, Zehui Chen,

Junjie Ye, Siyu Yuan, and 1 others. 2025. Critique-rl: Training language models for critiquing through two-stage reinforcement learning. *arXiv preprint arXiv:2510.24320*.

Zhihui Xie, Jie chen, Liyu Chen, Weichao Mao, Jingjing Xu, and Lingpeng Kong. 2025. Teaching language models to critique via reinforcement learning. In *ICLR 2025 Third Workshop on Deep Learning for Code*.

Zhangchen Xu, Yuetai Li, Fengqing Jiang, Bhaskar Ramasubramanian, Luyao Niu, Bill Yuchen Lin, and Radha Poovendran. 2025. Tinyv: Reducing false negatives in verification improves rl for llm reasoning. *arXiv preprint arXiv:2505.14625*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zhe Yang, Yichang Zhang, Yudong Wang, Ziyao Xu, Junyang Lin, and Zhifang Sui. 2025a. Confidence v.s. critique: A decomposition of self-correction capability for LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3998–4014, Vienna, Austria. Association for Computational Linguistics.

Zhe Yang, Yichang Zhang, Yudong Wang, Ziyao Xu, Junyang Lin, and Zhifang Sui. 2025b. A probabilistic inference scaling theory for LLM self-correction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13584–13598, Suzhou, China. Association for Computational Linguistics.

Zihuiwen Ye, Fraser David Greenlee, Max Bartolo, Phil Blunsom, Jon Ander Campos, and Matthias Gallé. 2025. Improving reward models with synthetic critiques. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4506–4520, Albuquerque, New Mexico. Association for Computational Linguistics.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025a. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Tianshu Yu, Chao Xiang, Mingchuan Yang, Pei Ke, Bosi Wen, Cunxiang Wang, Jiale Cheng, Li Zhang, Xinyu Mu, Chuxiong Sun, and Minlie Huang. 2025b. Training language model to critique for better refinement. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26760–26804, Vienna, Austria. Association for Computational Linguistics.

Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuewei Wang, Suchin Gururangan, Chao Zhang, Melanie Kambadur, Dhruv Mahajan, and Rui Hou. 2025c. Self-generated critiques boost reward modeling for language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11499–11514, Albuquerque, New Mexico. Association for Computational Linguistics.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, and 1 others. 2025. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Appendix

A More Implementation Details

We provide more implementation details in this section. We show the prompt utilized for cold start data distillation in Figure 6. We show some hyper-parameters in RL training in Table 3.

Prompt for Critique Data Distillation

#Question#:
<insert question>
#Solution#:
<insert question>
#Instruction#:
Please verify step by step and judge whether the solution is correct, and end your answer with **Conclusion: right/wrong [END]**

Figure 6: Our prompt fed to GPT-5 for critique data generation, which is comprised of a question, a solution, and critique instruction.

train bath size	512
ppo mini batch size	128
rollout n	5
adv estimator	grpo
kl loss coef	1e-3
learning rate	1e-6
max prompt length	4096
max response length	4096
clip ratio	0.2
epochs	10

Table 3: RL training hyper-parameters.

B More Experimental Results

We show more experimental results in this section. Main experiments on Mistral-7B-Instruct-v0.3 and Llama-3.1-8B-Instruct are shown in Table 4. Supplemental critique capability evaluation results are shown in Table 5. Performance comparison of static and dynamic critique training data is shown in Figure 7; performance comparison of different reward functions is shown in Figure 8; performance comparison of training with/without reward denoising strategy is shown in Figure 9.

C Discussions on Label Imbalance Problem

A theoretical analysis of data imbalance problem in critique training without the data filter. Theoretically, generating a critique can be viewed as a binary classification task, requiring the critique to include a final judgment on whether the response is correct or not, and binary classification tasks are susceptible to label distribution imbalance in the training data: when one category predominates, the trained model tends to favor predicting that category. In §3, the data processed through the first-order rollout and subsequently filtered by a data filter are stored in the *Question-response Data Cache* to support subsequent critique RL training, and this data filter ensures a 1:1 ratio between correct and incorrect responses, thereby preventing label bias in subsequent critique training. If the data filter is omitted and responses are sampled uniformly at random, in the first-order rollout, $E[R(r)] \times 100\%$ of responses are correct and $(1-E[R(r)]) \times 100\%$ are incorrect, where E denotes the mathematical expectation. Consequently, the ratio of correct to incorrect responses under random sampling is also $E[R(r)] : (1 - E[R(r)])$. Similar to Yang et al. (2025a), let P_1 and P_2 denote the probabilities that the model correctly identifies correct and wrong responses, respectively, during the critique phase. On a validation set with a balanced 1:1 positive-to-negative ratio, the expected validation reward is $E[R_{val}(c)] = \frac{0.7P_1+0.7P_2}{2}$, and the expected reward for critiques during RL training is:

$$\begin{aligned} & E[R(c)] \\ &= E[R(r)] * P_1 * 0.7 + (1 - E[R(r)]) * P_2 * 0.7 \\ &= 0.7E[R(r)]P_1 \\ &\quad + (1 - E[R(r)])(2E[R_{val}(c)] - 0.7P_1) \\ &= 0.7(2E[R(r)] - 1)P_1 \\ &\quad + 2(1 - E[R(r)])E[R_{val}(c)] \end{aligned} \tag{7}$$

Though $E[R(r)]$ and $E[R_{val}(c)]$ gradually improve throughout the whole RL training process, they can be approximately viewed as constants between adjacent RL steps. From this point of view and Equation 7, we can see that the expected critique reward is in proportion to P_1. Similar to Yang et al. (2025a), P_1 and P_2 also exhibit a competitive trade-off and $P_1 + P_2$ can be viewed as a constant between adjacent RL steps. When $2E[R(r)] - 1 > 0$, this reward encourages the model to increase P_1 and decrease P_2; when $2E[R(r)] - 1 < 0$, a

decrease in P_1 raises the expected reward, thereby incentivizing the model to reduce P_1 and increase P_2. For example, when training Qwen2.5-7B with GC-RL in §4, we find $E[R(r)] \approx 0.1$ in early training and $E[R(r)] \approx 0.45$ in late stage, consistently satisfying $2E[R(r)] - 1 < 0$ and incentivizing the model to decrease P_1. Consequently, a model trained with randomly sampled responses for second-order rollout exhibits lower P_1 and higher P_2, meaning it tends to classify responses as incorrect during critique.

The weighted reward function is unbiased We give a proof of the unbiasedness of the following weighted reward function:

$$R_w(c) = \begin{cases} \frac{0.35}{E[R(r)]}, & \textit{Ext(c) = correct \& R(r)=1} \\ \frac{0.35}{1-E[R(r)]}, & \textit{Ext(c) = wrong \& R(r)=0} \\ 0, & else \end{cases} \tag{8}$$

Under this scheme, the expected reward for critiques during RL training becomes:

$$\begin{aligned} & E[R(c)] \\ = & E[R(r)]P_1 \frac{0.35}{E[R(r)]} \\ & + (1 - E[R(r)])P_2 \frac{0.35}{1 - E[R(r)]} \\ = & \frac{0.7P_1 + 0.7P_2}{2} \\ = & E[R_{val}(c)] \end{aligned} \tag{9}$$

Under these circumstances, the reward does not incentivize the model to increase or decrease P_1 or P_2. Essentially, this weighting strategy amplifies the reward for rare classes, enabling the model to learn more effectively from such examples and thereby approximating balanced training.

Models	Methods	Math-500	GSM8k	Minerva	AMC23	Olympiad	Avg
		Generation Accuracy (%)					
Mistral-7B-Instruct-v0.3	w/o RL	10.8	46.7	7.4	5.0	1.8	14.3
	C-RL	19.8	54.2	9.8	12.5	4.2	20.1
	G-RL	47.6	77.8	18.6	30.0	15.6	37.9
	GC-RL	**52.1**	**81.2**	**19.4**	**32.5**	**17.9**	**40.6**
Llama-3.1-8B-Instruct	w/o RL	48.2	84.2	19.1	25.0	15.0	38.3
	C-RL	55.6	87.6	21.3	32.5	20.1	43.4
	G-RL	72.3	92.0	28.9	47.5	28.4	53.8
	GC-RL	**75.8**	**92.6**	**30.1**	**50.0**	**31.6**	**56.0**
		Critique Accuracy (%)					
Mistral-7B-Instruct-v0.3	C-RL	65.2	70.1	60.3	66.4	63.2	65.0
	GC-RL	**69.0**	**74.8**	**64.5**	**71.6**	**66.1**	**69.2**
Llama-3.1-8B-Instruct	C-RL	77.8	80.5	58.3	70.4	67.3	70.9
	GC-RL	**81.4**	**83.2**	**64.1**	**75.4**	**70.6**	**74.9**

Table 4: Generation and critique capabilities evaluation results on Mistral-7B and Llama3.1-8B-Instruct. GC-RL outperforms all other RL training methods in both generation and critique capabilities.

Models	Methods	Math-500	GSM8k	Minerva	AMC23	Olympiad	Avg
		Denoised Reward					
Qwen2.5-7B	C-RL	0.901	0.912	0.715	0.797	0.812	0.827
	GC-RL	0.945	0.986	0.774	0.852	0.836	0.879
Qwen2.5-3B	C-RL	0.752	0.754	0.652	0.699	0.691	0.710
	GC-RL	0.774	0.802	0.657	0.712	0.698	0.729
Qwen2.5-1.5B	C-RL	0.651	0.638	0.607	0.648	0.612	0.631
	GC-RL	0.683	0.676	0.658	0.672	0.635	0.665

Table 5: Supplemental critique capability evaluation results on Qwen-2.5-(1.5B,3B,7B). The average denoised reward (§5.2) is also reported to reflect the critique capability.

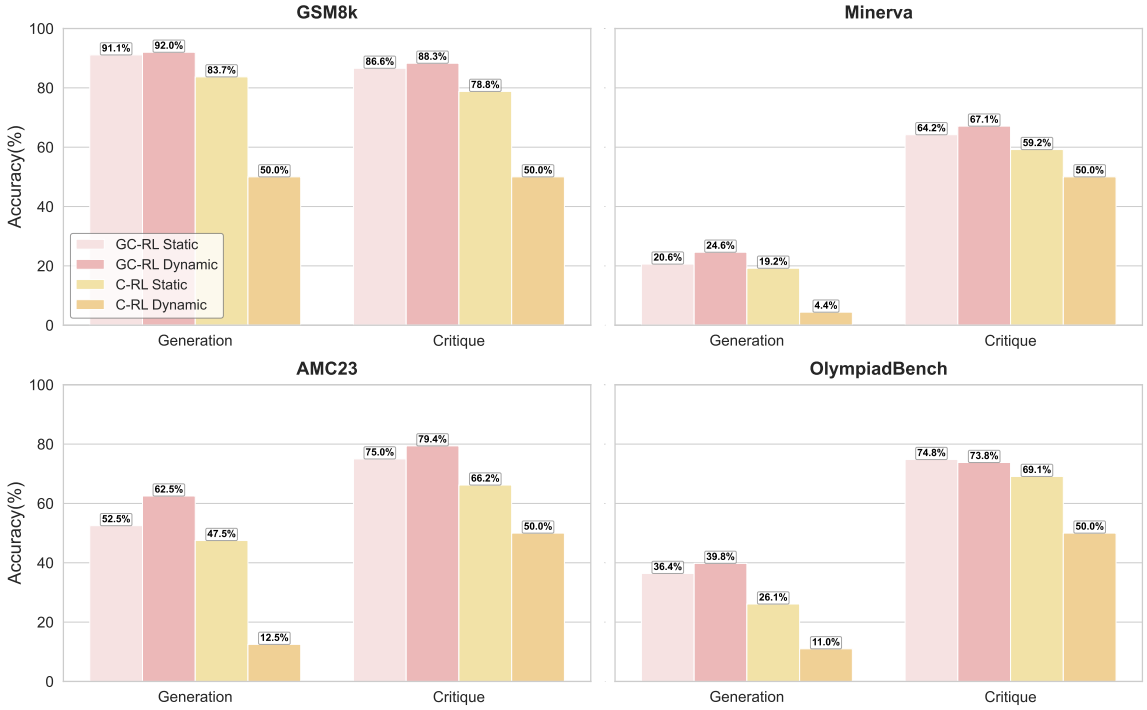

Figure 7: Performance of Qwen2.5-7B on 4 datasets under GC-RL and C-RL settings with static and dynamic critique training data. Dynamic data outperforms static data in the GC-RL setting, while the opposite holds for C-RL.

Figure 8: A comparison of critique performance of Qwen2.5-7B with different reward functions on 4 datasets. Compared to baseline $R(c)$, $R_w(c)$ leads to a higher precision while $R_r(c)$ generates a higher recall.

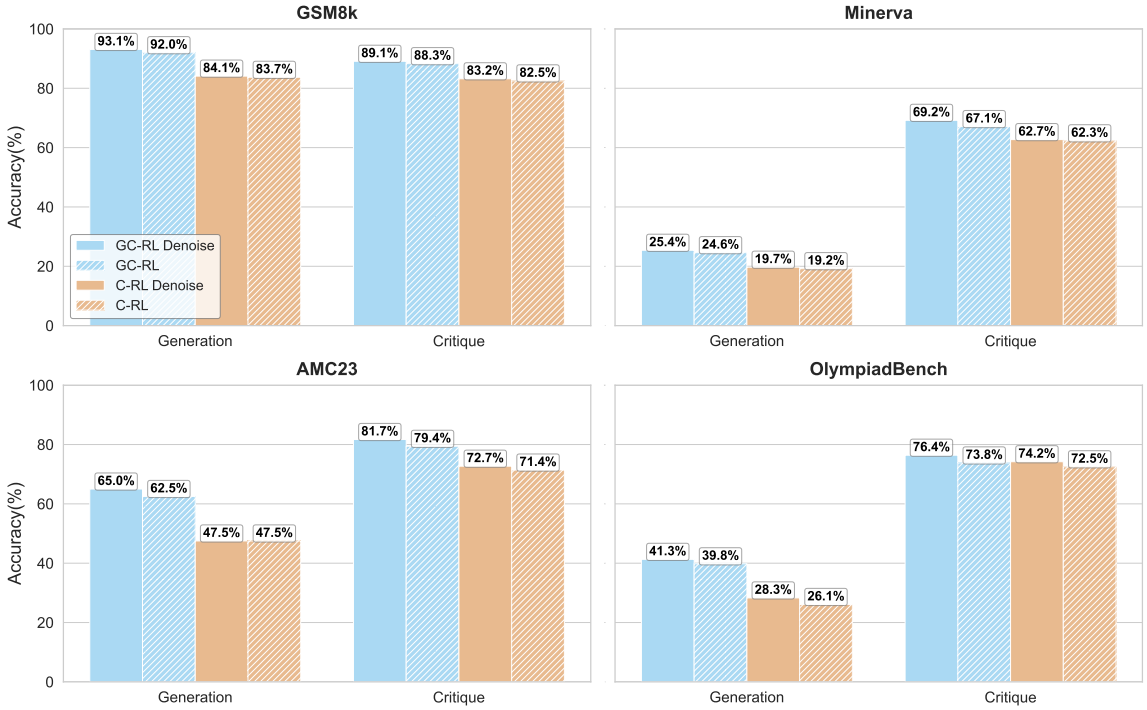

Figure 9: A comparison of model performance of Qwen2.5-7B with/without reward denoising strategy on 4 datasets. In both GC-RL and C-RL settings, reward denoising can improve model performance on both generation and critique capabilities.