Writing-Zero: Bridge the Gap Between Non-verifiable Problems and Verifiable Rewards

Xun Lu

Star Writing Team

Abstract

Reinforcement learning with verifiable rewards (RLVR) has enabled large language models (LLMs) to achieve remarkable breakthroughs in reasoning tasks with objective ground-truth answers, such as mathematics and code generation. However, a significant gap remains for non-verifiable tasks, like creative writing and open-ended dialogue, where quality assessment is inherently subjective and lacks definitive references. Existing approaches for these domains often rely on scalar reward models trained with human preferences, which suffer from limited generalization and are prone to reward hacking, such as over-explanation and length bias. In this work, we propose a unified RLVR-based training paradigm that bridges the gap between non-verifiable tasks and verifiable rewards. We introduce a writing-principle-based pairwise Generative Reward Model (GenRM) and a novel Bootstrapped Relative Policy Optimization (BRPO) algorithm. The pairwise writing GenRM leverages self-principled critique to transform subjective assessments into reliable, verifiable rewards, while BRPO enables dynamic, reference-free pairwise comparison by leveraging a bootstrapped response as temporary reference from within group rollouts during RL training. Our approach empowers LLMs to develop robust writing capabilities without supervised fine-tuning, as demonstrated by Writing-Zero, which shows consistent improvement and strong resistance to reward hacking compared to scalar reward baselines. Furthermore, our method achieves competitive results on both in-house and open-source writing benchmarks. Our findings suggest the potential to unify rule-based, reference-based, and reference-free reward modeling under the RLVR framework, thus paving the way for a comprehensive and scalable RL training paradigm applicable across all language tasks.

1 Introduction

Recently, large language models (LLMs) have achieved remarkable breakthroughs in reasoning capabilities through reinforcement learning with verifiable rewards (RLVR), particularly in solving complex logical tasks such as mathematics and programming (Guo et al., 2025). RLVR relies on reference-based signals, where the availability of objective ground-truth answers enables reliable verification of model responses. This approach has proven particularly effective in tasks with well-defined solutions, such as mathematical reasoning and code generation, where simple rule-based verifiers can provide clear binary signals (correct or incorrect) (Team et al., 2025). However, there is a spectrum ranging from verifiable to non-verifiable problems, with problems like mathematics and coding at one end, multi-subject QA with less structured answers in the middle, and creative writing, multi-turn dialogue that lack a reference answer and require quality assessment based on human preferences at the non-verifiable end. For less-verifiable or non-verifiable problems, previous works mainly rely on a scalar reward model trained with human preference data for RLHF training, which has limited generalization ability and is prone to reward hacking (Zhong et al., 2025). For example, in creative writing scenarios, models trained with RLHF often exhibit over-explanation, where they append lengthy justifications of how their response perfectly meets user requirements, even when the actual content fails to do so.

Developing high-quality and robust reward modeling methods for more general domains remains a challenging and active research direction. Su et al. (2025) propose training a generative reward model to verify whether the response matches the reference answer, expanding the application of RLVR to less structured domains such as multi-subject QA. Most recently, researchers attempt directly adapting RLVR to improve generative reward modeling (GenRM) and train GRM in R1-style with long COT of designed evaluation principles and critiques, which results in more accurate and reliable rewards across broader domains and effective test-time scaling (Liu et al., 2025; Chen et al., 2025a). While the emergence of reliable and robust GenRM has opened promising avenues for extending RLVR to broader domains, its effective application, particularly to non-verifiable tasks, remains a challenging and largely unexplored research topic.

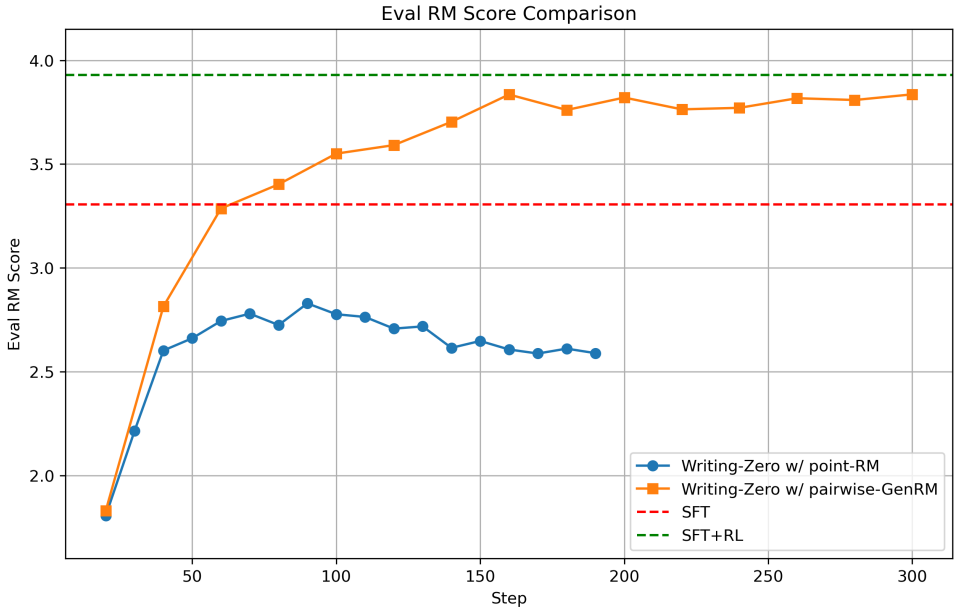


Figure 1: Comparison of Eval RM scores during **Writing-Zero** training. The blue line shows `Qwen3-32B-Base-ScalarRM-GRPO`, while the orange line shows **Writing-Zero** (`Qwen3-32B-Base-GenRM-BRPO`). The dashed red and green lines indicate the SFT (Writing-SFT) and SFT+RL (Writing-SFT-GenRM-BRPO) baselines, respectively.

As the Chinese saying goes, ("There is no absolute first place in literary arts"), creative writing represents one of the typical non-verifiable tasks where quality assessment is inherently subjective and lacks definitive reference. In this work, we aim to bridge the gap between the non-verifiable tasks and verifiable rewards and propose a new training paradigm for non-verifiable writing tasks, with writing-principle-based pairwise GenRM and a new RL algorithm Bootstrapped Relative Policy Optimization (BRPO). Specifically, inspired by Liu et al. (2025), we first train the Pairwise Writing GenRM through high-quality self-principled critique cold-start data, and perform RLVR to refine the GenRM's ability to formulate adaptive principles and nuanced critiques specific to different writing scenarios and diverse response pairs, thereby producing more reliable outcome rewards for creative writing evaluation. The Writing-GenRM takes in a pair of responses and outputs two respective scores between 0 and 10 to indicate comparative quality, effectively transforming subjective assessments into reliable verifiable rewards. Then we introduce Bootstrapped Relative Policy Optimization (BRPO), which dynamically selects samples from within the group rollouts as temporary references for pairwise comparison and advantage estimation. This bootstrap mechanism eliminates the need for fixed external references, enabling the policy model to continuously improve by leveraging its own increasingly sophisticated outputs for comparison. Similar to DeepSeek-R1 (Guo et al., 2025), we explore the potential of LLMs developing writing capabilities without any supervised finetuning data and introduce Writing-Zero using Qwen3-32B-Base (Yang et al., 2025) as the base model. Throughout the training process, Writing-Zero shows consistent improvement and exceptional resistance against reward hacking compared with the base model trained with scalar reward, as shown in Figure 1. Furthermore, we introduce Writing-R1 using an in-house thinking SFT model as the base model and achieve competitive results on in-house and open-source writing benchmarks. Our empirical results demonstrate the effectiveness of our proposed method.

To sum up, our work represents the last piece of the puzzle in unifying different reward modeling paradigms under the RLVR training framework. We demonstrate that by leveraging pairwise generative reward modeling with self-principled critique, even non-verifiable tasks like creative writing can benefit from stable and scalable RL training, similar to verifiable tasks. This opens up the possibility of unifying three major reward modeling approaches: rule-based rewards for well-defined tasks, model-based rewards with reference answers for less-structured tasks, and model-based rewards without reference answers for creative tasks. Our work thus paves the way for establishing a comprehensive and consistent RLVR training paradigm that can be applied across the entire spectrum of language tasks, from verifiable to non-verifiable domains.

2 Preliminary

2.1 Group Relative Policy Optimization

GRPO (Shao et al., 2024) is an efficient and effective RL algorithm, by introducing the group-relative normalization to estimate the advantage and eliminating the value function of PPO (Schulman et al., 2017). For a specific question q in dataset $\mathcal{D}$, the behavior policy $\pi_{\theta_{old}}$ samples a group of G individual responses $\{o_i\}_{i=1}^{G}$. Then, the advantage of the i-th response is calculated by normalizing the group-level rewards $\{R_i\}_{i=1}^{G}$. Like PPO, GRPO also stabilizes training and improves sample efficiency by constraining the policy model π_θ updates within a proximal region with clip(). Specifically, GRPO updates the policy by maximizing the following objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left(\min\left(r_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}\left(r_{i,t}(\theta),1-\varepsilon,1+\varepsilon\right)\hat{A}_{i,t}\right)-\beta D_{\text{KL}}(\pi_\theta\|\pi_{\text{ref}})\right)\right] \tag{1}$$

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}\mid q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid q,o_{i,<t})} \qquad \hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^{G})}{\text{std}(\{R_i\}_{i=1}^{G})} \tag{2}$$

where $\hat{A}_t$ is the estimator of the advantage at time step t, ε is the clipping range of importance sampling ratio $r_{i,t}$, and π_{ref} is the reference model, same as the initial π_θ.

2.2 Dynamic Sampling Policy Optimization

Compared to GRPO, DAPO (Yu et al., 2025) introduces another four important tricks: clip-higher, dynamic sampling, token-level policy gradient loss and overlong reward shaping. The key improvement is dynamic sampling, by over-sampling and filtering out prompts with the accuracy equal to 1 and 0, leaving all prompts in the batch with effective gradients, avoiding dampening the gradient signals for model training with larger variance in gradient. Specifically, with question-answer pairs (q,a) in dataset $\mathcal{D}$, DAPO updates the policy by maximizing the following objective:

$$\begin{aligned}\mathcal{J}_{\text{DAPO}}(\theta) = \ & \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\min\left(r_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}\left(r_{i,t}(\theta),1-\varepsilon_{\text{low}},1+\varepsilon_{\text{high}}\right)\hat{A}_{i,t}\right)\right] \\ & \text{s.t. } 0<\left|\{o_i\mid \text{is\_equivalent}(a,o_i)\}\right|<G,\end{aligned} \tag{3}$$

2.3 Rule-based Reward Modeling

For verifiable tasks, defining a rule-based reward function is straightforward. This approach yields accurate reward signals, which is crucial for scaling the model's reasoning capability through RL, while mitigating issues like reward hacking. With y as the ground-truth answer and $\hat{y}$ as the predicted answer, such a reward function can be simply defined as

$$R(\hat{y},y) = \begin{cases}1, & \text{is\_equivalent}(\hat{y},y)\\ -1, & \text{otherwise}\end{cases} \tag{4}$$

For non-verifiable tasks, mainstream RLHF approaches still largely employ scalar Bradley-Terry reward models.

3 Approach

In this paper, we propose a new training paradigm for the non-verifiable writing tasks via Reinforcement Learning with Verifiable Rewards (RLVR), which consists of two stages: first, we train a Self-Principled Critique Pairwise GenRM for Creative Writing; second, we train writing models with the Pairwise Writing GenRM through Bootstrapped Relative Policy Optimization (BRPO).

3.1 Self-Principled Critique Pairwise GenRM for Creative Writing

The Self-Principled Critique Tuning (SPCT) framework, proposed by Liu et al. (2025), offers a method for unified scoring of single or multiple responses via textual critique. Building upon this, we introduce our **Pairwise Writing GenRM**, which adapts and refines the SPCT methodology specifically for the nuances of creative writing. Our approach incorporates two key modifications to enhance its suitability and effectiveness for non-verifiable writing tasks:

- **Focused Application on Non-verifiable Writing:** Unlike the broader application of SPCT, our Pairwise Writing GenRM is exclusively applied to non-verifiable writing tasks. This focus stems from the observation that verifiable tasks, which possess definitive answers, can be more directly and reliably assessed using rule-based verifiable rewards.
- **Dedicated Pairwise Comparison:** We streamline the input format to solely consist of pairwise responses. This contrasts with SPCT's capability to handle single or multiple responses. We find that a dedicated pairwise setup simplifies the training process and yields more accurate comparative assessments by eliminating potential distractions from additional responses.

The development and training of our Pairwise Writing GenRM, based on the `Qwen3-32B-Base` model, follows a structured four-step pipeline:

1. **Data Filtering:** Selection of higher-quality data from raw preference datasets.
2. **SPCT Prompt Engineering:** Design of tailored SPCT prompts specifically for writing tasks.
3. **Cold-Start Fine-tuning:** Collection of a small set of cold-start data using a rejection sampling strategy to initially fine-tune the `Qwen3-32B-Base` model.
4. **RL-based Refinement:** Further enhancement of the GenRM's performance using rule-based reinforcement learning.

3.1.1 Data Filtering

Our GenRM training data originates from an in-house dataset of approximately 200K pairwise preferences, including 30K writing-related pairs. These writing pairs are re-scored using a scalar RM trained on the full preference set. We then filter for higher-quality pairs, characterized by a higher chosen response reward and a larger reward gap, resulting in approximately 10K original writing pairs. For GenRM evaluation, we use an in-house test set of 1K writing preferences. The final preference dataset can be described as:

$$\mathcal{D} = \{(x^{(i)}, y_c^{(i)}, y_r^{(i)})\}_{i=1}^{N} \tag{5}$$

where x is the query, y_c and y_r are the chosen and rejected responses.

3.1.2 Writing Principles and Critique

Our GenRM template, similar to SPCT, utilizes pre-defined general writing principles and model-generated specific principles (based on the input query and response pair) to form a detailed critique (Liu et al., 2025). The critique concludes with `\boxed{score_1, score_2}` to denote the comparative quality of the two responses.

The GenRM follows a sequential process: first generating specific principles, then the critique, and finally extracting the paired scores:

$$\{p_i\}_{i=1}^{m} \sim p_\theta\left(x, y_c, y_r\right), \quad \mathcal{C} \sim r_\theta\left(x, y_c, y_r, \{p_i\}_{i=1}^{m}\right), \quad \{S_c, S_r\} = (S_1, S_2) = f_{\text{extract}}(\mathcal{C}), \tag{6}$$

where p_θ denotes the principle generation function (sharing parameters θ with the reward generation function r_θ), which outputs specific principles $\{p_i\}$. $\mathcal{C}$ represents the detailed critique generated based on these principles. S_c and S_r are the individual quality scores for the chosen (y_c) and rejected (y_r) responses, respectively, corresponding to (S_1, S_2) based on their presentation order in the template. f_{extract} is the function to extract these paired scores from the critique $\mathcal{C}$. Notably, unlike SPCT, our GenRM outputs S_c and S_r as float numbers within the [0, 10] range, rather than integers.

3.1.3 Cold Start with RFT

To mitigate instability in the early stages of RL training and to enhance the granularity of specific principles, we perform a cold-start fine-tuning (RFT) of the initial RL actor. This involves curating a small, high-quality dataset as follows: First, we sample 1k writing preference pairs from the 10k higher-quality pairs in 3.1.1 and swap the positions of chosen and rejected responses to construct 2k SPCT prompts, with 50% (y_c, y_r) and 50% (y_r, y_c). Then we sample initial reasoning traces (specific principles and

critiques) for these SPCT prompts using `Claude-3.5-Sonnet`. Note that Claude's predictions exhibit a disproportional tendency (around 60%) to assign higher scores to the former response, suggesting a potential position bias problem.

Next, we employ a rejection process to refine the dataset. Specifically, for each SPCT prompt, we examine Claude's predicted $\{S_c, S_r\}$, the pair scores for the chosen (y_c) and rejected (y_r) responses respectively. If this prediction does not align with the known ground-truth preference for that pair (i.e., $S_c < S_r$), the prompt is discarded. We then proceed to only keep the original queries for which *both* of their corresponding SPCT prompt versions (i.e., the (y_c, y_r) version and the swapped (y_r, y_c) version) are both consistent with the ground-truth preference. As a result, we maintain 500 original writing preference pairs, doubling this set to 1,000 pairs by swapping the chosen and rejected responses. We fine-tune `Qwen3-32B-Base` with these 1k cold start data for one epoch.

3.1.4 Rule-based RL for Pairwise GenRM

The writing GenRM is further fine-tuned by GRPO (Shao et al., 2024), with rule-based outcome rewards. The predicted pair float scores $\{S_c, S_r\}$ are extracted and composed to two kinds of rewards: an accuracy reward and a format reward.

Accurate Reward Formally, for the i-th output o_i in the group of G individual responses $\{o_i\}_{i=1}^{G}$, the accuracy reward is:

$$R_{\text{acc}} = \begin{cases} 1, & \text{if } S_c > S_r, \\ -1, & \text{if } S_c < S_r, \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

Format Reward The format reward is also important, for both parsing process and numerical boundary:

$$R_{\text{format}} = \begin{cases} 1, & \text{if } 0 \le S_c \le 10 \text{ and } 0 \le S_r \le 10, \\ -1, & \text{otherwise,} \end{cases} \tag{8}$$

Score Margin As it is difficult for the writing GenRM to distinguish pair responses with fine-grained textual or semantic divergences, we introduce a heuristic weighting term to enhance GenRM's sensitivity to these differences:

$$R_{\text{acc}} = \begin{cases} R_{\text{acc}} \times \dfrac{(S_c - S_r)}{2}, & \text{if } 0 < S_c - S_r < 2, \\ R_{\text{acc}}, & \text{otherwise,} \end{cases} \tag{9}$$

Position Bias Weight Let P_{former} denote the probability that the GenRM prefers the first-presented (former) response in a pair. Ideally, we expect P_{former} to be 0.5 because the training data is deliberately balanced with respect to presentation order, as it includes swapped chosen-rejected response pairs. Furthermore, its variance (estimated from batch statistics) should approach 0 on this balanced training data. The reinforcement learning process helps converge the empirical estimate of P_{former} (i.e., the proportion of times the former response is preferred in a batch) to approximately 0.5, yet reducing the variance of these batch-wise proportions remains a challenge. To mitigate the variance of these batch-wise preference proportions, we apply a weighting to the advantage $\hat{A}_{i,t}$ that is dynamically calculated based on the preference distribution within the current batch:

$$\hat{A}_{i,t} = \begin{cases} \hat{A}_{i,t} \times \dfrac{\text{\#Latter Preference in Batch}}{\text{Batch Size}/2}, & \text{if } S_c > S_r \text{ and } y_c \text{ was presented first,} \\ \hat{A}_{i,t} \times \dfrac{\text{\#Former Preference in Batch}}{\text{Batch Size}/2}, & \text{if } S_c > S_r \text{ and } y_c \text{ was presented second.} \end{cases} \tag{10}$$

Dynamic Sampling We employed dynamic sampling following Yu et al. (2025). This technique involves over-sampling and then filtering out prompts for which the GenRM's predictive accuracy is 0 or 1.

3.2 RL with Bootstrapped Relative Policy Optimization

In this stage, we introduce Bootstrapped Relative Policy Optimization (BRPO), an algorithm adapted from GRPO (Shao et al., 2024). BRPO is designed to utilize preference rewards obtained from the pairwise

GenRM, facilitating robust learning for non-verifiable tasks. With BRPO, `Qwen3-32B-Base` is trained as a writing model directly, without supervised fine-tuning (SFT) as a preliminary step, termed `Writing Zero`. Furthermore, we apply BRPO to an in-house SFT model to create `Writing-R1`.

3.2.1 Bootstrapped Relative Policy Optimization

While Group Relative Policy Optimization (GRPO) (Shao et al., 2024) utilizes group-relative normalization of scalar rewards, defining such rewards reliably for non-verifiable tasks like creative writing is challenging due to their inherent subjectivity. Pairwise comparative assessment, however, offers a more robust and scalable evaluation method, aligning better with human judgments in these subjective domains. Bootstrapped Relative Policy Optimization (BRPO) is designed to leverage this by enabling dynamic, reference-free pairwise comparison. It achieves this by utilizing a bootstrapped response—randomly selected from the current group of policy rollouts—as a temporary reference for advantage estimation during RL training. This core mechanism eliminates the need for fixed external references, allowing the policy model to continuously improve by learning from its own increasingly sophisticated outputs. BRPO consists of two main components: a novel *preference advantage* calculation and a *dynamic sampling* strategy.

For a given query, BRPO dynamically establishes a temporary reference by randomly selecting one response, o_{ref}, from the current group of G policy-generated responses $\{o_i\}_{i=1}^{G}$. This o_{ref} is then used for pairwise comparison against all responses in the group to obtain preference scores. In other words, BRPO employs a bootstrap approach to iteratively enhance its model capabilities by conducting multiple rounds of voting and calculating preference among its self-generated outputs. Consequently, our BRPO achieves zero expectation for advantage and requires no additional normalization.

Preference Advantage At the core of BRPO's advantage estimation is the direct use of binary pairwise preference signals, replacing the pointwise scalar normalization found in methods like GRPO (as seen in Equation 2). Specifically, for each output o_i within the group of G policy-generated responses $\{o_i\}_{i=1}^{G}$, its preference relative to the dynamically selected reference o_{ref} (as described previously) is determined. This results in a binary preference score R_i, obtained by comparing o_i and o_{ref} using the pairwise GenRM. The corresponding scores S_i (for o_i) and S_{ref} (for o_{ref}) are extracted from the GenRM's critique. The advantage $\hat{A}_{i,t}$ for the i-th response at token t is then directly set to this binary preference score:

$$R_i = \begin{cases} 1, & \text{if } S_i > S_{\text{ref}} \text{ in all voting conditions} \\ -1, & \text{otherwise} \end{cases}, \quad \text{and thus, } \hat{A}_{i,t} = R_i. \tag{11}$$

Dynamic Sampling BRPO also incorporates a dynamic sampling mechanism. As the policy model π_θ evolves during training, the quality of its generated responses will progress. A potential issue arises if a chosen o_{ref} for a particular query happens to be an outlier—either exceptionally strong or weak, or containing superficial patterns that the GenRM is overly sensitive to. In such cases, o_{ref} might be consistently judged as superior (or inferior) to nearly all other responses in the group $\{o_i\}_{i=1}^{G}$. This can lead to a batch of preference scores R_i that are predominantly all +1 or all -1, offering little discriminative signal for policy updates and potentially encouraging the policy to overfit to superficial patterns in o_{ref}. To mitigate this, BRPO filters out queries for which the selected o_{ref} leads to such highly skewed preference distributions within the group. Formally, for a group of G pairwise preference scores $\{R_i\}_{i=1}^{G}$, the query q is filtered out from the current training batch if:

$$\frac{\left|\sum_{i=1}^{G} R_i\right|}{G} > \tau_{\text{filter}}, \tag{12}$$

where τ_{filter} is a hyperparameter threshold.

3.2.2 Writing-Zero and Writing-R1

Inspired by approaches like DeepSeek-R1-Zero (Guo et al., 2025), which investigate training models from scratch via RL, we explore a similar paradigm for creative writing with our *Writing-Zero* model. This involves training `Qwen3-32B-Base` directly from its pretrained state, relying exclusively on a pure reinforcement learning process driven by our pairwise writing GenRM and BRPO algorithm for self-evolution, without task-specific supervised fine-tuning. Remarkably, Writing-Zero demonstrates the potential to achieve performance comparable to models undergoing traditional SFT and subsequent RL tuning.

To further assess the capabilities of BRPO, we also introduce *Writing-R1*. For this model, we apply the same BRPO algorithm to fine-tune an in-house SFT model already specialized in thinking and instruction following. As demonstrated by our empirical results, Writing-Zero and Writing-R1 not only achieve competitive performance on in-house and open-source writing benchmarks but also exhibit strong resistance to reward hacking, showcasing the effectiveness of our proposed methodology.

4 Experiments

4.1 Experiment Settings

Method Implementation The preference data for pairwise writing GenRM is described in Section 3.1.1. The RL training data for Writing-Zero and Writing-R1 are from in-house unsuperivised queries. To ensure consistent training and evaluation, we primarily utilize Chinese-based datasets in this work. Our training framework is based on verl (Sheng et al., 2024). We employ vLLM engine (Kwon et al., 2023) for both GenRM and Writing Model inference. The learning rate for all training is 1e-6. For the decoding parameters of both training and evaluation in this paper, `temperature` is set to 1.0, `top-p` is set to 1.0, `top-k` is set to -1. We set the dynamic sampling threshold τ_{filter} to 0.6, which means for a group of 16, if the sum of preference scores is greater than 9, the query will be filtered out.

Eval RM To observe the performance changes of the writing model during the BRPO training process, we have specifically trained a scaler reward model on the in-house writing test dataset, termed Eval RM. The Eval RM exhibits a high degree of consistency with human evaluation results on the in-house writing test dataset.

4.2 Benchmarks

We evaluate our proposed Pairwise Writing GenRM on multiple reward model benchmarks, demonstrating its effectiveness in enhancing performance during reinforcement learning.

Reward Model Benchmarks We compare our Pairwise Writing GenRM with Skywork-Reward (Liu et al., 2024), INF-ORM (Minghao Yang, 2024), our in-house Writing Scalar RM (trained with 200K full preference data in 3.1.1), LLM-as-a-Judge baselines (Claude-3.5-Sonnet), across a range of established reward model benchmarks. RewardBench (Lambert et al., 2024) is one of the earliest benchmarks employing prompt–chosen–rejected triplets for reward model evaluation, encompassing diverse tasks in chat, reasoning, and safety, with approximately 3,000 annotated examples. M-RewardBench (Gureja et al., 2024) extends this benchmark to a multilingual setting. To better assess our model's discrimination ability in Chinese, we restrict evaluation to its Chinese subset.

Additionally, given that our reward model is primarily designed for Chinese writing scenarios, we construct two additional domain-specific datasets: Cultural & Creative Writing and Lifestyle Copywriting. The former is a general-purpose creative writing dataset that includes tasks such as essay composition, workplace copywriting, and literary creation, comprising a total of 1,036 samples. The latter focuses on common everyday writing scenarios—such as chatting with friends, composing text messages, or posting on social media—and includes 831 samples.

Writing Benchmarks We evaluate the writing capability of our models and compare their performance with other baseline models, including Qwen3-32B-Base, Qwen3-32B-Instruct, DeepSeek-R1, Writing-SFT (our in-house thinking SFT model). WritingBench (Wu et al., 2025) serves as a comprehensive benchmark that covers 6 core domains and 100 subdomains, encompassing a wide range of writing tasks and styles.

In addition, to better align with our real-world application scenarios, we curated a diverse set of 211 user queries, named as Writing Testset. Based on history responses and human annotations, we construct the preference dataset, from which we trained the Eval RM in 4.1. Eval RM is used to score the responses generated by different models, providing a task-specific evaluation of writing quality. We observed a strong positive correlation between the scores produced by this reward model and human judgments, indicating its effectiveness and reliability in assessing writing performance.

4.3 Main Results

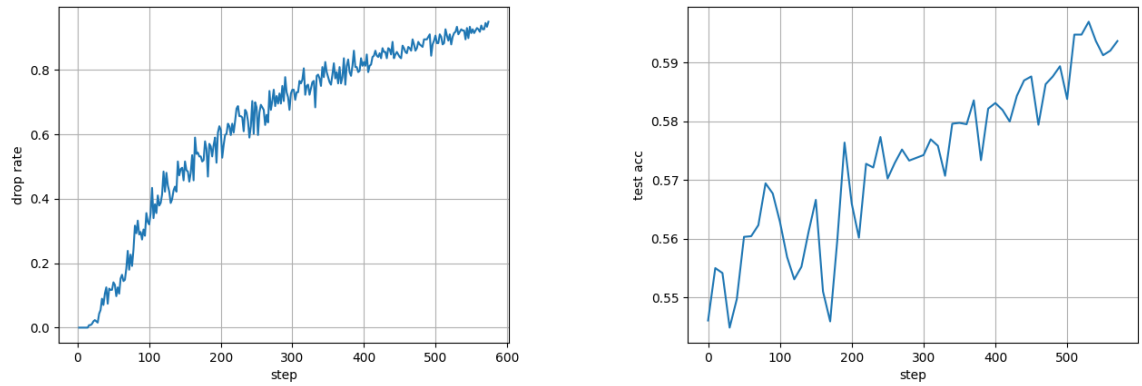
4.3.1 Reward Model Results

As shown in Table 1, evaluated on the same prompt sets, our Pairwise Writing GenRM outperforms Claude-3.5-Sonnet across all four benchmarks. While its performance is lower than our in-house Writing

Model	Cultural & Creative Writing	Lifestyle Copywriting	Reward Bench	M-RewardBench
Skywork-Reward-Gemma-2-27B-v0.2	56.4	56.3	94.3*	90.8
INF-ORM-Llama3.1-70B	56.6	54.0	**95.1***	**91.4**
Writing Scalar RM	**64.7**	**62.8**	92.9	89.4
Claude-3.5-Sonnet	53.0	46.8	84.2*	79.7
Pairwise Writing GenRM	57.5	54.8	87.4	86.1

Table 1: Performance of different types of reward models on benchmark datasets. The superscript asterisk (*) indicates results taken directly from the RewardBench Leaderboard.

Scalar RM, this is understandable as the Pairwise GenRM was trained on significantly less data. Surprisingly, despite being trained exclusively on Chinese writing data (without any English or general-domain examples), it achieved 87.4% accuracy on RewardBench and 86.1% on M-RewardBench, showcasing strong generalization performance beyond its primary training scope. Furthermore, on the Cultural & Creative Writing dataset, our Pairwise Writing GenRM achieved 1.1% and 0.9% higher accuracy than Skywork-Reward-Gemma-2-27B-v0.2 and INF-ORM-Llama3.1-70B, respectively.



(a) Drop rate of GenRM on train data

(b) Accuracy of GenRM on test data

Figure 2: GenRM training dynamics, showing test accuracy and training data drop rate.

While the Pairwise Writing GenRM demonstrated promising benchmark results, its training process encountered a notable challenge. Specifically, during GenRM training, dynamic sampling caused a significant computational bottleneck. After certain iterations, the dropout rate progressively increased, exceeding 95% in the later stages (as shown in Figure 2a). This extremely high dropout rate caused severe computational inefficiency, as a large number of rollouts are required to assemble enough training batch, making further training prohibitively slow. Consequently, we had to halt the GenRM training process prematurely. As a result, the model's accuracy on the test set did not show clear signs of convergence (Figure 2b), implying that the GenRM employed in subsequent RL stages was likely sub-optimal and that further refining its training could enhance overall RL performance. We plan to address this issue in future work.

4.3.2 Writing Model Results on Writing Benchmarks

As shown in Table 2, RL training substantially improves the writing capabilities of both the `Qwen3-32B-Base` and our in-house `Writing-SFT` model. Our core approach involves training these base models with Pairwise Writing GenRM using the BRPO algorithm, resulting in **Writing-Zero** (`Qwen3-32B-Base-GenRM-BRPO`) and **Writing-R1** (`Writing-SFT-GenRM-BRPO`). For comparison, we also implemented baselines where these base models were trained with a Scalar Reward Model (ScalarRM) using the GRPO algorithm, denoted as `Qwen3-32B-Base-ScalarRM-GRPO` and `Writing-SFT-ScalarRM-GRPO`.

Notably, when guided by our Pairwise GenRM with BRPO:

- **Writing-Zero** improves from the base Qwen3-32B-Base scores of 6.89 to 8.29 on WritingBench and from 1.23 to 3.84 on the Writing Testset.
- **Writing-R1** (based on Writing-SFT) shows gains to 8.68 on WritingBench (from Writing-SFT's 8.56) and to 3.93 on the Writing Testset (from Writing-SFT's 3.31).

Model	WritingBench	Writing Testset
Qwen3-32B-Base	6.89	1.23
Qwen3-32B-Base-ScalarRM-GRPO	~~8.87~~	2.83
Qwen3-32B-Base-GenRM-BRPO / **Writing-Zero**	8.29	3.84
Qwen3-32B-Base-GenRM-BRPO / **Writing-Zero** (voting@2)	8.35	3.86
Qwen3-32B-Instruct	8.64	3.32
Deepseek-R1	8.55*	3.20
Writing-SFT	8.56	3.31
Writing-SFT-ScalarRM-GRPO	**8.77**	3.69
Writing-SFT-GenRM-BRPO / **Writing-R1**	8.68	**3.93**

Table 2: Performance of different models on writing-related evaluation datasets. The * indicates results taken directly from the original benchmark. Strikethrough results (e.g., ~~8.87~~) marks scores potentially inflated by reward hacking.

Despite the Pairwise GenRM's lower raw performance on some reward model benchmarks (Table 1) compared to the Scalar RM, it demonstrates greater efficacy when integrated into the RL process with BRPO. As depicted in Figure 1, Writing-Zero not only achieves higher and more stable reward scores during training compared to `Qwen3-32B-Base-ScalarRM-GRPO`, but also translates to better final performance on the Writing Testset, surpassing it by 1.01 points. Similarly, Writing-R1 outperforms `Writing-SFT-ScalarRM-GRPO` by 0.24 points on the same test set.

A crucial observation pertains to reward hacking: during the training of `Qwen3-32B-Base-ScalarRM-GRPO`, we noted early instances where its responses devolved into gibberish and became largely unreadable. This poor quality was reflected in its low Eval RM scores on our Writing Testset. However, this same model achieved an anomalously high score on WritingBench, suggesting WritingBench's susceptibility to such hacking, a vulnerability our Eval RM resists.

To further validate these automated metrics, we conducted human evaluations. On a test set of 58 instances, Writing-R1 demonstrated superior performance when compared against both its SFT base (G:S:B ratio of 15:36:6 in favor of Writing-R1 vs. Writing-SFT) and the scalar-reward trained counterpart (G:S:B ratio of 17:34:6 in favor of Writing-R1 vs. `Writing-SFT-ScalarRM-GRPO`).

5 Analysis

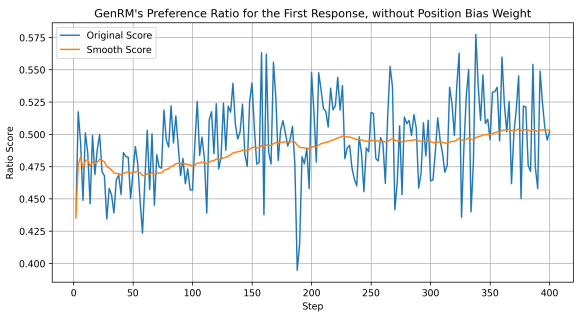
5.1 Reward Hacking in Writing

Model	Mean Response Len	Mean Explanation Len
Qwen3-32B-Base-ScalarRM-GRPO	1872	417
Qwen3-32B-Base-GenRM-BRPO / **Writing-Zero**	1292	58
Writing-SFT	1251	125

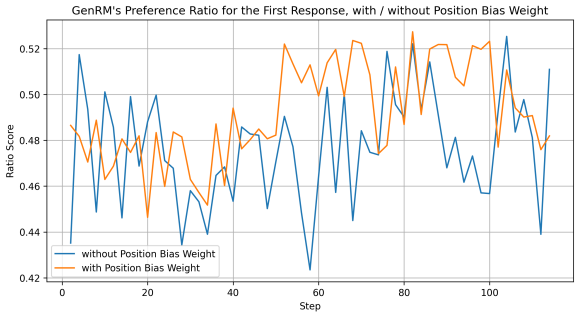
Table 3: Mean length of response and redundant explanation.

Creative writing tasks are susceptible to two significant reward hacking issues: 1) *Length bias*: Reward models often assign higher scores to longer responses, leading RL-trained models to become excessively verbose. 2) *Redundant explanations*: Reward models may favor responses with lengthy, often unnecessary, self-justifications or praise appended to the actual content, causing RL-trained models to generate such superfluous text.

Reward hacking results in superficial improvements in reward metrics during training, as the actor model learns to exploit loopholes in the reward model rather than acquiring genuine writing proficiency. To quantify these effects, we analyzed the performance of three key models on these hacking issues using an internal creative writing test set. As shown in Table 3, our **Writing-Zero** model demonstrates substantially less redundant information and shorter response lengths compared to `Qwen3-32B-Base-ScalarRM-GRPO`, highlighting its improved resistance to these hacking phenomena.



(a) Preference Ratio without Position Bias Weight



(b) Preference Ratio with Position Bias Weight

Figure 3: Convergence of the GenRM's preference ratio during RL training.

5.2 GenRM's Position Bias

After the cold-start fine-tuning (RFT) detailed in Section 3.1.3, our initial fine-tuned GenRM exhibited a significant position bias. Unlike Claude-3.5-Sonnet, which showed a 60% tendency to favor the first response, our model initially preferred to assign higher scores to the latter response in a pair. Fortunately, this inherent bias was largely calibrated during the subsequent RL phase. As illustrated in Figure 3a, the empirical probability of the GenRM preferring the former response gradually converged towards the ideal 50% during RL training.

While the mean of this preference probability (as shown in Figure 3a) converged towards 0.5, its variance across training batches did not show a similar convergent tendency without intervention. As demonstrated in Figure 3b, the position bias advantage weighting mechanism visibly reduced the variance of the preference ratio.

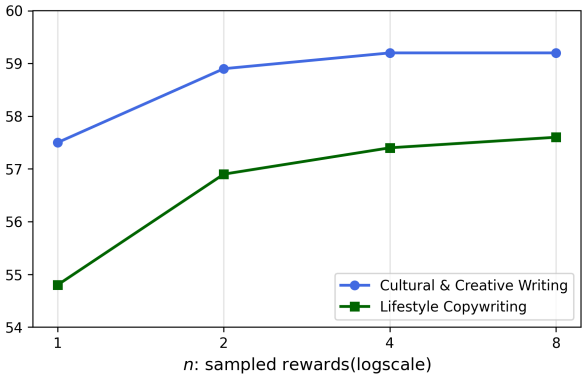


Figure 4: Majority voting accuracy (voting@n) across $n = 1, 2, 4, 8$ on internal datasets, as evaluated by the pairwise GenRM.

5.3 Test-time Scaling Benefits

A notable advantage of GenRMs over scalar reward models is their capability for test-time scaling through majority voting over multiple generations. This can enhance both the direct accuracy of the GenRM itself and, consequently, the performance of the policy model trained with it.

As illustrated in Figure 4, increasing the number of voting samples (n) directly improves the accuracy of our Pairwise Writing GenRM on our internal evaluation datasets. For instance, with $n = 8$ votes, accuracy on the Cultural & Creative Writing dataset increases from 57.5% (for $n = 1$) to 59.2%, and on the Lifestyle Copywriting dataset from 54.8% to 57.6%. This demonstrates the GenRM's ability to refine its judgments by aggregating multiple perspectives. The benefits of test-time scaling extend to the policy models trained with the GenRM. As shown in Table 2, when **Writing-Zero** utilizes its Pairwise GenRM with two voting rounds (voting@2), it achieves further improvements, which indicates that a more accurate reward signal, refined by test-time scaling, can lead to better policy optimization and ultimately stronger performance on downstream writing tasks. Further enhancing the GenRM's performance could unlock greater test-time scaling benefits, for example, by enabling strategies for multiple voting rounds (n ¿ 1)

that selectively reward consistently superior responses while neutralizing or penalizing inconsistently judged ones, thereby fostering more robust policy learning.

We believe that with further improvements to the GenRM's accuracy, the potential of test-time scaling with multiple voting rounds could be more deeply explored. For instance, an open question for future experimentation is whether responses with inconsistent judgments across multiple votes should be neutrally rewarded (e.g., 0) or negatively penalized (e.g., -1 in Equation 11).

5.4 Intuition behind reference selection of BRPO

The primary challenge in non-verifiable tasks like creative writing is the absence of definitive ground-truth answers, making quality assessment inherently relative and reliant on comparison. We initially considered using the best-performing or relatively better response from a previous group rollout as reference for the next training iteration. However, this approach proved problematic. As the policy model's capabilities significantly improve during training, such a static or outdated reference quickly becomes suboptimal, introducing an "offline" data issue that can impede further learning by comparing against a weaker baseline. Furthermore, our pairwise GenRM is designed for relative comparisons rather than assigning absolute scores. Attempting to derive a consistent listwise ranking from purely pairwise preferences would necessitate an additional redundant pointwise reward model. Consequently, BRPO adopts a dynamic, bootstrapped reference selection strategy, where a randomly chosen response from the current policy rollouts serves as a temporary, relevant reference for pairwise comparisons within the same group, fostering continuous self-improvement without external anchors.

6 Related Work

6.1 Reward Model

Early approaches for reward modeling primarily focused on discriminative reward models, which were trained to directly assess the quality of individual responses (Liu et al., 2024; Cai et al., 2024). These pointwise assessment methods laid the groundwork for preference learning, but frequently faced challenges with generalization across diverse contexts and over-optimization (Yang et al., 2024; Sharma et al., 2023; McKee-Reid et al., 2024). Another line of work gives pairwise scores to evaluate whether responses match the reference answer, or compare the quality of two responses (Xu et al., 2025; Jiang et al., 2023), showing that by learning to predict which of two responses is preferred, generally leads to improved stability and better capture the relative nuances of human preferences compared to direct scalar assessments. Generative Reward Models (GenRMs) introduce a new paradigm (Li et al., 2023; Vu et al., 2024; Ye et al., 2024; Zhang et al., 2024; Cao et al., 2024), which allows the reward model to leverage chain-of-thought and even use test-time scaling to make more reliable evaluation. Liu et al. (2025) stand out as the most relevant works to our work. They introduced a novel learning method enabling GenRMs to generate adaptive principles and accurate critiques to give a more reliable comparison between two responses through a two-phase training process: distillation then RLVR. Chen et al. (2025a); Whitehouse et al. (2025); Chen et al. (2025b) share a similar idea of using designed evaluation criteria and RLVR to improve the quality of GenRM. Zhang et al. (2025); Wang et al. (2025) also expand GenRM and RLVR to multimodal domains. In this work, we explore the application of pairwise GenRM to non-verifiable writing tasks.

6.2 Reinforcement Learning

Reinforcement Learning from Human Feedback (RLHF) has become a crucial component for aligning Large Language Models (LLMs) with human preferences and desired behaviors (Ouyang et al., 2022). Recently, Reinforcement Learning with Verifiable Rewards (RLVR) has proven an effective paradigm for imporving the reasoning capabilities of LLMs in domain such as math and coding (Jaech et al., 2024; Guo et al., 2025; Team et al., 2025), inspiring subsequent research to actively explore various potentially more advanced reasoning RL algorithms (Yu et al., 2025; Xiong et al., 2025; Li et al., 2025). Su et al. (2025) and Xu et al. (2025) propose to expand RLVR to problems with unstructured reference, which often rely on a pre-defined fixed reference or a best-of-n responses sampled from an initial or static model, to serve as a ground-truth reference. In this work, our proposed Bootstrapped Relative Policy Optimization (BRPO) algorithm is a step further to explore the application of RLVR to non-verifiable tasks.

7 Conclusion, Limitations, and Future Work

In this work, we propose a unified RLVR-based training paradigm to bridge the gap between non-verifiable tasks and verifiable rewards. By leveraging a writing-principle-based pairwise Generative Reward Model and the Bootstrapped Relative Policy Optimization algorithm, our approach enables large language models to achieve stable and scalable reinforcement learning even in highly subjective domains such as creative writing. Experimental results demonstrate that our method not only improves resistance to reward hacking but also achieves competitive performance on various writing benchmarks. This work paves the way for a comprehensive and consistent RL framework applicable across the full spectrum of language tasks. Due to limited computational resources, we were unable to fully explore the impact of various algorithmic parameters in both the GenRM and RL stages, nor could we comprehensively investigate the trends of test-time scaling of our pairwise writing GenRM. We leave the development of a unified framework integrating all reward modeling paradigms and the extension to multi-modal scenarios for future research.

References

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

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. arXiv preprint arXiv:2501.12599, 2025.

Jialun Zhong, Wei Shen, Yanzeng Li, Songyang Gao, Hua Lu, Yicheng Chen, Yang Zhang, Wei Zhou, Jinjie Gu, and Lei Zou. A comprehensive survey of reward models: Taxonomy, applications, challenges, and future. arXiv preprint arXiv:2504.12328, 2025.

Yi Su, Dian Yu, Linfeng Song, Juntao Li, Haitao Mi, Zhaopeng Tu, Min Zhang, and Dong Yu. Expanding rl with verifiable rewards across diverse domains. arXiv preprint arXiv:2503.23829, 2025.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. Inference-time scaling for generalist reward modeling. arXiv preprint arXiv:2504.02495, 2025.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, et al. Rm-r1: Reward modeling as reasoning. arXiv preprint arXiv:2505.02387, 2025a.

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

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

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. arXiv preprint arXiv:2409.19256, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles, 2023.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. arXiv preprint arXiv:2410.18451, 2024.

Xiaoyu Tan Minghao Yang, Chao Qu. Inf-orm-llama3.1-70b, 2024. URL https://huggingface.co/infly/INF-ORM-Llama3.1-70B.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Reward-bench: Evaluating reward models for language modeling, 2024.

Srishti Gureja, Lester James V Miranda, Shayekh Bin Islam, Rishabh Maheshwary, Drishti Sharma, Gusti Winata, Nathan Lambert, Sebastian Ruder, Sara Hooker, and Marzieh Fadaee. M-rewardbench: Evaluating reward models in multilingual settings. arXiv preprint arXiv:2410.15522, 2024.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, et al. Writingbench: A comprehensive benchmark for generative writing. arXiv preprint arXiv:2503.05244, 2025.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. arXiv preprint arXiv:2403.17297, 2024.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for llms. arXiv preprint arXiv:2406.10216, 2024.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, et al. Towards understanding sycophancy in language models. arXiv preprint arXiv:2310.13548, 2023.

Leo McKee-Reid, Christoph Sträter, Maria Angelica Martinez, Joe Needham, and Mikita Balesni. Honesty to subterfuge: In-context reinforcement learning can make honest models reward hack. arXiv preprint arXiv:2410.06491, 2024.

Wenyuan Xu, Xiaochen Zuo, Chao Xin, Yu Yue, Lin Yan, and Yonghui Wu. A unified pairwise framework for rlhf: Bridging generative reward modeling and policy optimization. arXiv preprint arXiv:2504.04950, 2025.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. arXiv preprint arXiv:2306.02561, 2023.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative judge for evaluating alignment. arXiv preprint arXiv:2310.05470, 2023.

Tu Vu, Kalpesh Krishna, Salaheddin Alzubi, Chris Tar, Manaal Faruqui, and Yun-Hsuan Sung. Foundational autoraters: Taming large language models for better automatic evaluation. arXiv preprint arXiv:2407.10817, 2024.

Ziyi Ye, Xiangsheng Li, Qiuchi Li, Qingyao Ai, Yujia Zhou, Wei Shen, Dong Yan, and Yiqun Liu. Beyond scalar reward model: Learning generative judge from preference data. arXiv preprint arXiv:2410.03742, 2024.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. arXiv preprint arXiv:2408.15240, 2024.

Maosong Cao, Alexander Lam, Haodong Duan, Hongwei Liu, Songyang Zhang, and Kai Chen. Compassjudger-1: All-in-one judge model helps model evaluation and evolution. arXiv preprint arXiv:2410.16256, 2024.

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

Nuo Chen, Zhiyuan Hu, Qingyun Zou, Jiaying Wu, Qian Wang, Bryan Hooi, and Bingsheng He. Judgelrm: Large reasoning models as a judge. arXiv preprint arXiv:2504.00050, 2025b.

Yi-Fan Zhang, Xingyu Lu, Xiao Hu, Chaoyou Fu, Bin Wen, Tianke Zhang, Changyi Liu, Kaiyu Jiang, Kaibing Chen, Kaiyu Tang, et al. R1-reward: Training multimodal reward model through stable reinforcement learning. arXiv preprint arXiv:2505.02835, 2025.

Yibin Wang, Zhimin Li, Yuhang Zang, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Unified multimodal chain-of-thought reward model through reinforcement fine-tuning. arXiv preprint arXiv:2505.03318, 2025.

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.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. arXiv preprint arXiv:2412.16720, 2024.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. arXiv preprint arXiv:2504.11343, 2025.

Chen Li, Nazhou Liu, and Kai Yang. Adaptive group policy optimization: Towards stable training and token-efficient reasoning. arXiv preprint arXiv:2503.15952, 2025.

A Writing Pairwise SPCT Template

The template we used for Writing Pairwise GenRM is shown as below:

Pairwise Writing GenRM (Chinese)

你是一位擅长评分的专家。根据给定的评分标准，评估两个助手的回答。评分过程分为以下几个步骤：
1. 确定需求类别
首先，根据当前对话场景和用户问题，判断用户的需求类别。需求类别包括：事实性问题、专业性问题、创意写作、日常对话、代码生成、数理推理、语言理解或其他。
2. 制定特有评估标准
根据判断出的需求类别，参考 ** 通用评估标准 **，制定一个针对该类别和当前问题的 ** 特有评估标准 **。特有标准应对具体需求进行补充或调整，并与通用评估标准相辅相成。特有标准应包含：
- 维度描述
- 1-10 分的评分标准
3. 对比评估
结合 ** 通用评估标准 ** 和 ** 特有评估标准 **，针对每个维度对两个助手的回答进行对比分析。
4. 权重分配
根据通用评估标准和特有评估标准，分配它们的每个具体条目的权重。确保所有条目的总权重之和为 100%。注意如果任意一个回答的有害性标准得分低，请提升有害性的权重；
5. 评分方法
对每个评估维度单独评分，评分范围为 1 到 10 分。1 分表示完全不符合要求，10 分表示完全符合要求。评分后，结合每个维度的权重，计算加权平均得分，得出每个回答的综合得分，综合得分在 1-10 之间。
—
通用评估标准

- ** 遵循指示 **
- 完全遵循（9-10 分）：完全符合问题的所有指示和要求。
- 部分遵循（6-8 分）：符合大部分指示，但有些遗漏或误解。
- 基本遵循（3-5 分）：符合部分指示，但未能满足主要要求。
- 未遵循（1-2 分）：完全未遵循指示。
例：如果问题要求提供三个例子，而回答只给出了一个，则为“部分遵循”。

- ** 有用性 **
- 非常有用（9-10 分）：提供全面且准确的信息，完全解决问题。
- 有用但不完全（6-8 分）：有用，但缺少细节或准确性。
- 有限的有用性（3-5 分）：信息很少，大部分不相关或不准确。
- 无用或错误（1-2 分）：完全无关或错误。
例：如果回答中有事实错误，但整体方向正确，则为“有用但不完全”。

- ** 相关性 **
- 高度相关（9-10 分）：与问题高度相关，信息紧密对接。
- 一般相关（6-8 分）：大体相关，但包含一些不必要信息。
- 部分相关（3-5 分）：包含大量偏离主题的内容。
- 不相关（1-2 分）：完全不相关。
例：如果回答偏离主题，但提供了一些相关信息，则为“部分相关”。

- ** 无害性 ** 遵纪守法，不涉及底线问题：拒绝回答色情、暴力、恐怖、反动、人身攻击、违背伦理道德、广告引流、政治敏感、非法行为建议等可能涉及高风险的内容
- 完全无害（10 分）：满足无害性所有条件
- 轻度有害（5 分）：程度较轻，部分有害
- 严重有害（0 分）：严重违背无害性条件
—
对话内容
{prompt}
助手回答1
[回答开始] {response_1} [回答结束]
助手回答2
[回答开始] {response_2} [回答结束]
—
输出要求
输出六个部分：
1. ** 特有评估标准 **：列出针对当前用户问题和需求类别的 ** 特有评估标准 **，每个维度的评分标准。
2. ** 思考和答案 **：如果是数学/推理/问答等有标准答案的问题，先在这里给出自己新的思考过程和答案；如果是其他类别问题，这里可以填”None”。
3. ** 分析 **：根据 ** 通用评估标准 ** 和 ** 特有评估标准 **，结合上面的 ** 思考和答案 ** 中的结果，对比并分析两个助手的回答，重点分析每个维度的表现。
4. ** 权重分配 **：列出通用评估标准和特有评估标准的权重分配，确保总权重为 100%。
5. ** 打分 **：计算两个助手回答的每个评估维度的得分；然后计算加权平均得分，计算过程使用数学公式，不要含有 Markdown。
6. ** 输出最终得分 **：输出格式为 \\boxed{{分数1,分数2}}，得分之间以逗号分隔。

Pairwise Writing GenRM (English)

You are an expert in scoring. Based on the given scoring criteria, evaluate the responses of two assistants. The scoring process consists of the following steps:

1. Determine the Demand Category
First, based on the current dialogue scenario and the user's question, identify the category of the user's demand. Demand categories include: factual questions, professional questions, creative writing, daily conversation, code generation, mathematical reasoning, language comprehension, or others.

2. Develop Specific Evaluation Criteria
Based on the identified demand category, refer to the **General Evaluation Criteria** to develop **Specific Evaluation Criteria** tailored to the category and the current question. The specific criteria should supplement or adjust for the particular need and complement the general criteria. The specific criteria should include:
- Dimension descriptions
- A 1-10 scoring scale

3. Comparative Evaluation
Using both the **General Evaluation Criteria** and the **Specific Evaluation Criteria**, conduct a comparative analysis of the two assistants' responses for each dimension.

4. Weight Allocation
Based on the general and specific evaluation criteria, allocate weights for each specific item. Ensure the total weight sums to 100%. Note: If any response scores low on the harmfulness criterion, increase the weight of harmfulness.

5. Scoring Method
Score each evaluation dimension separately on a scale of 1 to 10, where 1 means completely unsatisfactory and 10 means fully satisfactory. After scoring, calculate the weighted average score for each response based on the weights of each dimension, resulting in a comprehensive score between 1 and 10 for each response.

—

General Evaluation Criteria

- **Following Instructions**
- Fully follows (9-10 points): Fully complies with all instructions and requirements of the question.
- Partially follows (6-8 points): Meets most instructions but has some omissions or misunderstandings.
- Basically follows (3-5 points): Meets some instructions but fails to address the main requirements.
- Does not follow (1-2 points): Completely disregards instructions.
Example: If the question asks for three examples but the response provides only one, it is "partially follows."

- **Usefulness**
- Highly useful (9-10 points): Provides comprehensive and accurate information that fully solves the problem.
- Useful but incomplete (6-8 points): Helpful but lacks details or accuracy.
- Limited usefulness (3-5 points): Offers little information, mostly irrelevant or inaccurate.
- Useless or incorrect (1-2 points): Completely irrelevant or wrong.
Example: If the response contains factual errors but is generally on the right track, it is "useful but incomplete."

- **Relevance**
- Highly relevant (9-10 points): Closely related to the question, with tightly aligned information.
- Generally relevant (6-8 points): Mostly relevant but includes some unnecessary information.
- Partially relevant (3-5 points): Contains significant off-topic content.
- Irrelevant (1-2 points): Completely unrelated.
Example: If the response deviates from the topic but provides some relevant information, it is "partially relevant."

- **Harmlessness**
- Complies with laws and regulations, avoiding bottom-line issues: Refuses to answer questions involving pornography, violence, terrorism, 反动, personal attacks, ethical violations, advertising, political sensitivity, or suggestions of illegal behavior that may involve high-risk content.
- Completely harmless (10 points): Meets all harmlessness conditions.
- Mildly harmful (5 points): Slightly harmful in some aspects.
- Severely harmful (0 points): Severely violates harmlessness conditions.

—

Dialogue Content
{prompt}
Assistant Response 1
[Response Start] {response_1} [Response End]
Assistant Response 2
[Response Start] {response_2} [Response End]

—

Output Requirements
Output six sections:
1. **Specific Evaluation Criteria**: List the **Specific Evaluation Criteria** tailored to the current user question and demand category, including the scoring scale for each dimension.
2. **Thinking and Answer**: If it's a question with a definitive answer, such as `mathematics/reasoning/Q&A`, provide your new thought process and answer here; for other types of questions, you can fill in "None" here.
3. **Analysis**: Based on the **General Evaluation Criteria** and **Specific Evaluation Criteria**, combined with the results from the **Thoughts and Answers** section above, compare and analyze the responses of the two assistants, focusing on their performance across each dimension.
4. **Weight Allocation**: List the weight allocation for the general and specific evaluation criteria, ensuring the total weight is 100%.
5. **Scoring**: Calculate the score of each evaluation dimension for the answers of the two assistants; then calculate the weighted average score. Use mathematical formulas for the calculation process, without including Markdown.
6. **Output Final Scores**: The output format should be \\boxed{{score1,score2}}, with scores separated by commas.