Writing-RL: Advancing Long-form Writing via Adaptive Curriculum Reinforcement Learning

Anonymous ACL submission

Abstract

Recent advances in Large Language Models (LLMs) have enabled strong performance in long-form writing, but current training paradigms remain limited: Supervised Fine-Tuning (SFT) remains constrained by data saturation and performance ceilings, while Reinforcement Learning with Verifiable Reward (RLVR), though successful in verifiable domains like math and code, cannot be directly migrated to open-ended long-form writing due to a lack of ground-truths. To further advance long-form writing, we present **Writing-RL**: an *Adaptive Curriculum Reinforcement Learning* framework to advance long-form writing capabilities beyond SFT. The framework consists of three key components: *Margin-aware Data Selection* strategy that prioritizes samples with high learning potential, *Pairwise Comparison Reward* mechanism that provides discriminative learning signals in the absence of verifiable rewards, and *Dynamic Reference Scheduling* approach, which plays a critical role by adaptively adjusting task difficulty based on evolving model performance. Experiments on 7B-scale writer models show that Writing-RL effectively improves long-form writing performance over strong SFT baselines. Furthermore, we observe that models trained with long-output RL generalize surprisingly well to long-input reasoning tasks, potentially offering a promising perspective for rethinking long-context training.

1 Introduction

Recent years have witnessed the remarkable advance of Large Language Models (LLMs) (OpenAI, 2023; DeepSeek-AI et al., 2025; Zhao et al., 2023) to follow complicated instructions and provide helpful responses. Among their impressive capabilities, long-form writing, which aims to generate long and high-quality articles, has drawn increasing attention (Wu et al., 2025b; Bai et al., 2024b; Wu et al., 2025c) due to its broad practical applications.

However, generating articles of both sufficient length and high quality is non-trivial for current LLMs. Previous research has identified several challenges to employ LLMs for long-form generation, including inherently limited output ceiling (Bai et al., 2024b; Tu et al., 2025) and performance degradation as output length grows (Wu et al., 2025c; Tu et al., 2025). To address these issues, recent efforts perform targeted Supervised Fine-Tuning (SFT) on LLMs to extend their output lengths, with long-generation datasets constructed by iterative agent pipelines (Bai et al., 2024b; Quan et al., 2024; Wu et al., 2025c) or instruction back-translation (Pham et al., 2024; Wang et al., 2024). Though effective, these approaches introduce heavy burdens of dataset construction due to the broad coverage of writing tasks and potential copyright issues (Maini et al., 2024) when incorporating human-written texts. Furthermore, training LLMs to imitate the collected long-generation responses inherently imposes a capability upper bound determined by teacher models or human experts, which may cause data saturation and sample inefficiency.

Meanwhile, recent progress of Reinforcement Learning (RL) with Verifiable Rewards (DeepSeek-AI et al., 2025; Team et al., 2025; Yuan et al., 2025) in reasoning-intensive areas reveals a promising direction to advance model capabilities beyond SFT. In long-form writing, however, the lack of ground truths prevents a straightforward transfer of these successes. Wu et al. (2025a) utilize static reward models for grading, failing to dynamically adapt to evolving model capability. Overall, adaptive online RL for long-form writing remains under-explored and presents several challenges:

- **Data Selection**: Data quality and difficulty play a critical role in eliciting model potential. However, the optimal approach for selecting data

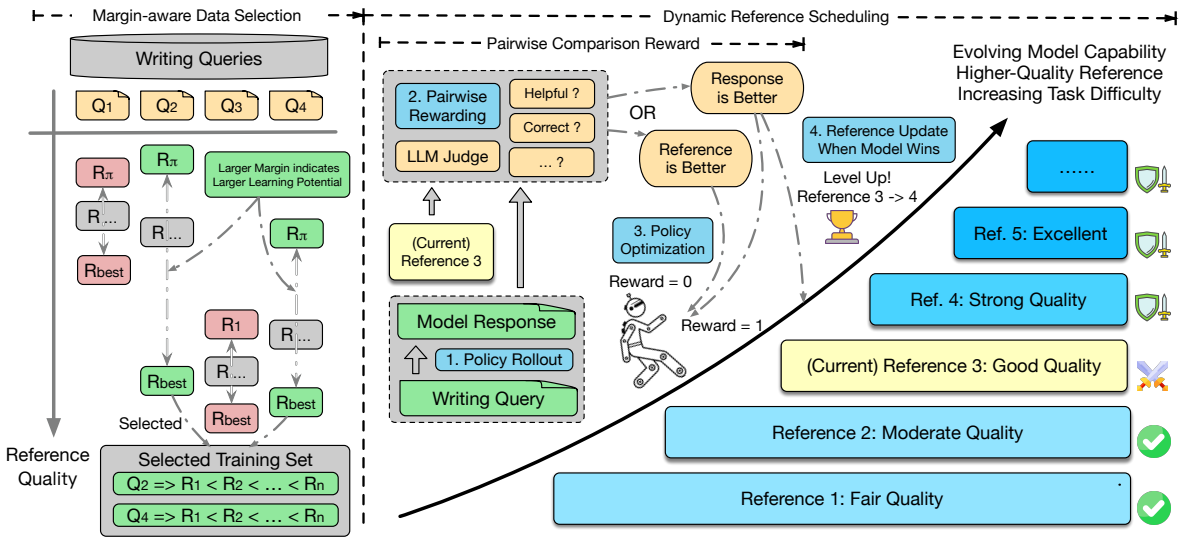

Figure 1: Overall framework of **Writing-RL**. 1) *Margin-aware Data Selection*: prioritizes samples with high learning potential; 2) *Pairwise Comparison Reward*: provides more discriminative reward signals; 3) *Dynamic Reference Scheduling*: adaptively incentivizes the model to surpass progressively stronger references.

for RL in long-form writing tasks remains unclear, requiring more explorations towards better learning efficiency.

- **Reward Design**: Rule-based outcome rewards (DeepSeek-AI et al., 2025) cannot be directly applied to generative writing tasks. Without ground-truth labels, constructing an effective reward mechanism for long-form writing poses a significant challenge.
- **Curriculum Scheduling**: Curriculum Learning (Bengio et al., 2009) is widely used to progressively improve model performance, but current static scheduling fails to adapt to the model's evolving competence, thereby reducing training effectiveness.

To tackle these challenges, our work proposes **Writing-RL**: an *Adaptive Curriculum Reinforcement Learning* framework tailored for long-form writing. As illustrated in Figure 1, our framework begins with ***Margin-aware Data Selection*** strategy which leverages the quality differential between the policy model response and the highest-quality reference as a measure of *learning potential*, diverging from the conventional difficulty-prioritized selection approach. Considering the limited discriminative capacity of pointwise rewarding, we construct a ***Pairwise Comparison Reward*** mechanism which challenges the policy model to generate responses of better quality than provided references to earn positive rewards. To facilitate progressive model enhancement, we propose a ***Dynamic Reference Scheduling*** approach that assigns each query a set of references with progressively increasing quality. The scheduling approach dynamically updates the references per sample when the evolving policy model surpasses the current reference during training. In this way, the dynamic curriculum adjusts sample-level task difficulty based on the current model performance, encouraging the model to consistently outperform a marginally superior reference. This rationale aligns with insights from recent R1-like RL practices (Shi et al., 2025; Bae et al., 2025) that samples neither too easy nor too difficult help to achieve the best learning efficiency.

To evaluate our framework, we conduct continuous reinforcement training on top of supervised fine-tuned writer models. The results indicate that our RL framework effectively boosts the long-form writing capability, advancing the SOTA performances of 7B-level writer models. Besides the improvement in long-form generation, we also observe an inspiring generalization phenomenon: our RL-trained writer model *(average input length < 1k)* shows a surprising improvement in long-text reasoning tasks *(input length: 8k–2M)*, in contrast to the performance degradation of the SFT-trained model. The results suggest a novel perspective on long-context learning: models trained on *long-output* tasks may also improve their reasoning abilities on *long-input* tasks, offering new insights into the relationship between long-context understanding and generation.

In summary, the contributions of our work are:

- We propose **Writing-RL**: an *Adaptive Curricu-*

lum Reinforcement Learning framework for long-form writing, which integrates three key components: *Margin-aware Data Selection*, *Pairwise Comparison Reward*, and *Dynamic Reference Scheduling*.

- Particularly, we propose **Dynamic Reference Scheduling**, which adaptively adjusts sample-level task difficulty based on the model's evolving performance. This dynamic curriculum encourages the model to continually outperform progressively stronger references.
- Our writer model achieves state-of-the-art performance at its scale, demonstrating the effectiveness of Writing-RL. Furthermore, we observe inspiring **Output-to-Input Generalization** from *long-output* generation to *long-input* reasoning, potentially offering a novel perspeactive to rethink long-context training.

2 Related Work

Training Methods for Long-form Writing. Recent efforts to advance long-form writing capabilities (Bai et al., 2024b; Wu et al., 2025c) mainly focuses on constructing long-generation datasets for fine-tuning. Main approaches include teacher model distillation (Wu et al., 2025c), iterative agent pipelines for extended output (Bai et al., 2024b; Tu et al., 2025; Quan et al., 2024) and instruction back-translation (Pham et al., 2024; Wang et al., 2024). (Wu et al., 2025a) incorporates static reward models for supervision, which fails to dynamically adapt to evolving model capability during training. However, the application of adaptive online reinforcement learning methods are relatively underexplored, hindering further improvement.

Long-form Writing Evaluation. Long-form writing (Wu et al., 2025b) requires LLMs to write open-ended articles, posing challenges for evaluation due to the lack of ground-truths. Earlier studies establish writing benchmarks (Wu et al., 2025c; Que et al., 2024), with proprietary models (Bai et al., 2024b; Paech, 2023; Liu et al., 2024) or fine-tuned LLMs (Wu et al., 2025c; Ke et al., 2024) to serve as judges. However, there exist several biases of including position bias and self-enhancement bias (Zheng et al., 2023), challenging the reliability of LLM-as-Judge evaluation methods.

Curriculum Learning. Reinforcement Learning methods (Schulman et al., 2017; Shao et al., 2024; DeepSeek-AI et al., 2025) have become a critical step to elicit LLM capabilities. To boost efficiency, Curriculum Learning (Bengio et al., 2009) has been widely adopted in RL practices (Team et al., 2025; Xie et al., 2025; Wen et al., 2025), including static difficulty-based scheduling (Luo et al., 2025; Song et al., 2025) and dynamic data selection (Bae et al., 2025; Shi et al., 2025). However, they use rule-based correctness as a measure for difficulty and perform sample selection, which increases rollouts and may cause imbalanced learning across samples.

3 Writing-RL

In this work, we propose **Writing-RL**, an *Adaptive Curriculum Reinforcement Learning* framework aimed at further improving long-form writing capabilities. The framework comprises three key components: *Margin-aware Data Selection*, *Pairwise Comparison Reward* and *Dynamic Reference Scheduling*. We will describe the components in detail respectively.

3.1 Margin-aware Data Selection

Previous data selection approaches typically take question difficulty as a key criteria, measured by the accuracy of the policy model (Shi et al., 2025; Bae et al., 2025), simplistic indicators (Cheng et al., 2021; Yang et al., 2025) like solution step counts or simple heuristics grounded in human intuition (Hendrycks et al., 2021b). While difficulty-prioritized data selection has been effective in verifiable tasks such as math and code, it depends on clearly defined ground truth to measure difficulty. In open-ended writing tasks, however, the lack of ground-truths makes difficulty hard to measure.

To address this issue, we propose *Margin-aware Data Selection*, which uses the performance gap between the policy output and the highest-quality reference as a measure of *learning potential*. Our intuition is simple: a question suitable for learning is a question with sufficient room for performance improvement. Specifically, the procedure is detailed as follows.

Generation with Multiple LLMs. Instead of relying on a single model as the difficulty estimator (Shi et al., 2025; Bae et al., 2025), we leverage a set of competitive LLMs $\mathcal{C} = \{\pi, M_1, M_2, \dots\}$, including the policy model π, to generate diverse candidate responses for each writing instruction.

Multi-dimensional Grading. Each generated response r_j from model $M_j \in \mathcal{C}$ is graded using a multi-dimensional pointwise LLM-as-a-Judge approach (Liu et al., 2024; Wu et al., 2025c), with averaged quality score denoted as s_j per response.

Data Selection on Learning Potential. To prioritize samples from which the policy model can benefit most, we define the *model-grounded learning potential* p as the quality gap between the best competitor and the policy model:

$$p = \max_{j \in \mathcal{C},\, j \neq \pi} (s_j - s_\pi)$$

where s_π is the score of the policy model's response. A higher p indicates greater headroom for improvement. To filter out potential noise, we first discard samples where all the competitors receives low scores, as such instructions are often overly difficult or suffer from quality issues themselves. After filtering, we retain the top-k examples based on their learning potential p to construct the training set.

3.2 Pairwise Comparison Reward Mechanism

Reward function is a critical component to guide policy optimization in RL. While rule-based outcome reward (DeepSeek-AI et al., 2025; Team et al., 2025) has been proven to be remarkably effective in verifiable tasks, it can not be directly applied to long-form writing due to the lack of ground-truths and its subjective nature, posing challenges to reward design.

Recent efforts utilize LLM-as-a-Judge (Zheng et al., 2023; Wu et al., 2025c) to measure the quality of model-generated responses. There exists two evaluation approaches including pointwise grading and pairwise comparison. Though widely adopted due to its simplicity, pointwise grading exhibits limited discriminative capabilities and relatively high variance. On the contrary, pairwise comparison evaluates the response against a high-quality reference, capturing the subtle differences and potential direction of improvement. By providing more discriminative reward signals, pairwise comparison incentivizes the model to generate better response and defeat high-quality references for positive rewards. Therefore, our reward design is as follows:

$$r_{\text{quality}}(\mathbf{x}) = \begin{cases} 1 & \text{if Judge}(\mathbf{ref}, \mathbf{x}) = \mathbf{x} \succ \mathbf{ref} \\ 0.5 & \text{if Judge}(\mathbf{ref}, \mathbf{x}) = \mathbf{x} \equiv \mathbf{ref} \\ 0 & \text{if Judge}(\mathbf{ref}, \mathbf{x}) = \mathbf{x} \prec \mathbf{ref} \end{cases}$$

where $r_{\text{quality}}(\mathbf{x})$ denotes the reward for a generated response $\mathbf{x}$; $\mathbf{ref}$ represents the high-quality reference response; and $\text{Judge}(\mathbf{ref}, \mathbf{x})$ is the evaluation function performed by the LLM-based judge to compare $\mathbf{x}$ with $\mathbf{ref}$.

In our experiments detailed in Appendix D.1, our model judge achieves a relatively high agreement (0.75) with human judges. Furthermore, LLM judges are known to exhibit position bias (Zheng et al., 2023) in pairwise comparisons, systematically favoring the first response. To impose additional learning pressure, we deliberately place the model-generated response in the second position, thereby introducing *positional disadvantage* in training. This avoids the need for position-swapped comparisons and halves the evaluation cost, while encouraging the model to generate stronger outputs from a less favorable position.

3.3 Dynamic Reference Scheduling

Curriculum Learning (Bengio et al., 2009) schedules progressive task difficulty for better learning efficiency. Previous efforts utilize offline-calculated difficulty for scheduling (Shi et al., 2025; Song et al., 2025) or introducing additional rollouts during training for adaptive sample selection (Bae et al., 2025; Yu et al., 2025). Though effective in reasoning-centered RL, these methods suffer from either non-adaptive difficulty estimates or increased inference overhead.

To address this issue, we propose a *Dynamic Reference Scheduling* approach that encourages the policy model to sequentially outperform references of ascending quality. With the algorithm detailed in Algorithm 1, our framework introduces a more competitive reference when the policy model beats the current one during training, enabling asynchronous per-sample difficulty updates and dynamic adaptivity with the evolving model.

Prior to Training: Data Preparation. Given a set of writing instructions W, we first apply the Margin-aware Data Selection strategy as elaborated in Sec 3.1, obtaining multiple competitive references $\mathcal{R} = \{r_\pi, r_1, r_2, \dots\}$ and their corresponding LLM-judged quality scores $\mathcal{S} = \{s_\pi, s_1, s_2, \dots\}$ for each instruction. The references are then sorted in ascending order of quality to produce a stage-wise reference list $\mathcal{R}_s = \{r_{q1}, r_{q2}, \dots\}$. To maintain sufficient positive feedback early in training, we deliberately include the response from the initial policy model π in the reference set, as the other reference-generation LLMs are generally more competent.

During Training: Dynamic Scheduling. At the start of training, each instruction is initialized with the lowest-quality reference r_{q1}, which is comparable to the initial policy model's response. As the model evolves during training, the model gradually generates higher-quality responses during rollouts and receives positive rewards in some of the pair-

Algorithm 1 Dynamic Reference Scheduling for Long-form Writing

Pre-processing: For each instruction $w \in W$, apply Margin-aware Data Selection (Section 3.1) to obtain a stage-wise reference list $\mathcal{R}^{(w)} = \{r_\pi^{(w)}, r_1^{(w)}, r_2^{(w)}, \dots\}$ ordered by ascending quality.
Input: Instruction set W; reference lists $\{\mathcal{R}^{(w)}\}_{w \in W}$; policy model π_θ; RL updater $\mathcal{A}$ (e.g., PPO); batch size B.
Initialize reference pointer $t_w \leftarrow 1$ **for all** $w \in W$ ▷ current reference index
while training not finished **do**
  Sample batch $\mathcal{B} = \{w_k\}_{k=1}^{B}$ from W
  for all $w_k \in \mathcal{B}$ **do**
    $r_k \leftarrow \mathcal{R}^{(w_k)}[t_{w_k}]$ ▷ current reference
    Generate response $g_k \leftarrow \pi_\theta(w_k)$
    Compute reward $R_k \leftarrow \mathrm{Judge}(r_k, g_k)$ ▷ 1 (win), 0.5 (tie), 0 (loss)
  end for
  Update policy $\pi_\theta \leftarrow \mathcal{A}\big(\pi_\theta, \{(w_k, g_k, R_k)\}_{k=1}^{B}\big)$
  for all $w_k \in \mathcal{B}$ **such that** $R_k = 1$ **do** ▷ reference surpassed
    if $t_{w_k} < |\mathcal{R}^{(w_k)}|$ **then**
      $t_{w_k} \leftarrow t_{w_k} + 1$ ▷ promote to next stronger reference
    end if
  end for
end while

wise comparisons. Subsequently, the defeated references r_t are replaced with marginally stronger ones r_{t+1} while the undefeated references are retained, progressively increasing the challenge without overwhelming the model, in alignment with the model's evolving capability. This dynamic and adaptive reference update mechanism establishes an asynchronous learning schedule for each writing instruction and effectively incentivize the model to consistently perform better.

4 Experiments

To demonstrate the effectiveness of Writing-RL, we conduct experiments on writing-oriented fine-tuned LLMs to see whether it can further advance long-form writing capabilities beyond SFT.

4.1 Datasets

We use two carefully-constructed writing datasets primarily designed for SFT, including LongWriter training set (Bai et al., 2024b) and WritingBench training set (Wu et al., 2025c). As detailed in Section 3.1, we perform the *Margin-aware Data Selection* on the datasets respectively. Specifically, we first generate references for each writing instruction with the initial policy model and four competent LLMs, including Qwen-Plus (Yang et al., 2024), GPT-4o (Hurst et al., 2024), Claude-3.7 (Anthropic Team, 2025) and DeepSeek-R1 (DeepSeek-AI et al., 2025). Then, we utilize a fine-tuned judge model (Wu et al., 2025c), which is optimized for evaluating long-form writing responses, to grade the responses in multiple dimensions. Finally, after the selection process, we obtain 1.5k chosen samples each dataset for further reinforcement learning. Each sample contains a writing instruction and references ordered by ascending quality. Additionally, we present analysis of topic coverage in Appendix B.1.

4.2 Training Setup

To fully harness the full potential of reinforcement learning, we use two writing-expert LLMs as the base models for RL, which are primarily fine-tuned with the full WritingBench training set, denoted as *Qwen2.5-7B-WritingBench-SFT* and *Llama3.1-8B-WritingBench-SFT* respectively.

With the proposed Writing-RL, we use the PPO algorithm (Schulman et al., 2017) to optimize the models for long-form writing. During the training process, we adopt Qwen-Plus to serve as pairwise-comparison judge, providing rewards for policy optimization. We include more details about reward model choice and training parameters in Appendix A.3 and Appendix A repesctively.

4.3 Benchmarks and Baselines

To comprehensively evaluate long-form writing capabilities of LLMs, we use three established benchmarks including WritingBench (Wu et al., 2025c), LongBench-Write (Bai et al., 2024b), and Creative-Writing-Bench (EQ-Bench creative writing split) (Paech, 2023). The benchmarks are of broad coverage and use strong judge LLMs to evaluate the quality of generated responses. We conduct three independent runs and report average performance for both model families to reduce variance. Note that the judge LLMs adopted for evaluation are diverse and different from the rewarding judge LLM used in training, mitigating the risk of overfitting particular judge preferences to ensure a

Model	Writing-Oriented Training		Long-form Writing Evaluation			
	SFT	RL	WritingBench	Creative-W.B.	LongBench-Write	Average
(a) Proprietary LLMs						
Qwen-Plus	–	–	77.62	76.78	95.42	83.27
GPT-4o	–	–	83.42	80.45	92.92	85.60
(b) Writing-Oriented Fine-Tuned LLMs						
Suri-7B	✓	✗	49.70	18.44	33.44	33.86
Longwriter-9B	✓	DPO	79.10	44.15	80.83	68.03
Longwriter-Zero-32B	✓	GRPO	82.92	61.14	85.90	76.65
(c) Qwen2.5-7B-Instruct Model Family						
Qwen2.5-7B-Instruct	✗	✗	73.16	49.29	87.20	69.88
Qwen2.5-7B-WritingBench-SFT (12k)	✓	✗	83.71	70.24	92.56	82.17
Qwen2.5-7B-WritingBench-SFT (24k)	✓	✗	83.76	69.60	92.65	82.00
Qwen2.5-7B-Reference-SFT	✓	✗	84.23	68.89	92.88	82.00
Qwen2.5-7B-Writing-RL (Ours)	✓	PPO	**87.23**	**73.19**	**93.06**	**84.49**
(d) Llama3.1-8B-Instruct Model Family						
Llama3.1-8B-Instruct	✗	✗	66.56	48.83	80.79	65.39
Llama3.1-8B-WritingBench-SFT	✓	✗	83.75	77.70	90.67	84.04
Llama3.1-8B-Reference-SFT	✓	✗	83.98	76.70	91.53	84.07
Llama3.1-8B-Writing-RL (Ours)	✓	PPO	**87.11**	**82.67**	**92.09**	**87.29**

Table 1: Evaluation results of the models trained with Writing-RL, with the highest score in each model family **bold**. Notably, our trained models perform the best within their model family, on par with the proprietary models.

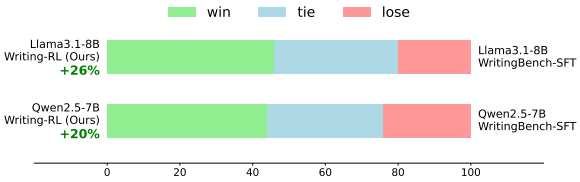

Figure 2: **Human Evaluation** results of pairwise comparison between our RL-trained models and the best-performing SFT-trained competitors.

fair evaluation.

Our selected baselines include strong proprietary models, instruction fine-tuned LLMs (Yang et al., 2024; Dubey et al., 2024), writing-oriented fine-tuned LLMs (Wu et al., 2025c; Bai et al., 2024b; Pham et al., 2024; Wu et al., 2025a), and the models supervisedly fine-tuned on our RL dataset. More evaluation details can be found in Appendix B.

4.4 Results

Remarkable Writing Performance Gains. As shown in Table 1, the results demonstrate that our models outperform others across the benchmarks. Specifically, *Llama3.1-8B-Writing-RL* achieves the highest average score of 87.29, and *Qwen2.5-7B-Writing-RL* follows with an average of 84.49, both showing strong performance at their scale. Notably, our trained models exhibit long-form writing capabilities that match or even surpass those of proprietary models, positioning them as strong open-source alternatives for long-form generation.

SFT Encounters Data Saturation. We observe distinct performance trends when applying RL and SFT to relatively strong models. Despite using identically constructed datasets, we observe a marginal performance regression in *Qwen2.5-7B-WritingBench-SFT (24k)* relative to its counterpart *Qwen2.5-7B-WritingBench-SFT (12k)*. Furthermore, the models continuously fine-tuned with high-quality references in the RL dataset, namely *Llama3.1-8B-Reference-SFT* and *Qwen2.5-7B-Reference-SFT*, also show minimal performance gain, or even slight degradation. This observation potentially underscores the phenomenon of data saturation, where as the model gets stronger, it is harder to construct SFT data to effectively further improve model capability.

RL Leads to Further Improvements. In contrast, models trained by Writing-RL demonstrate consistent improvements and thereby indicates the promising potential of RL to further advance model capability where SFT encounters limitations.

4.5 Human Evaluation

Furthermore, we recognize that human evaluation could serve as a great supplement to automatic LLM-as-Judge. Therefore, we also conduct human evaluation experiments to further validate our model performance. We randomly sample 100 writing instructions in total from our evaluation datasets and generate responses using our RL-trained models and the most competitive base-

Model	Writing-Oriented Training		Evaluation					
	SFT	RL	Easy	Hard	Short	Medium	Long	Overall
Qwen2.5-7B-Instruct	✗	✗	31.8	28.3	38.9	**26.0**	21.3	29.6
Qwen2.5-7B-WritingBench-SFT	✓	✗	27.6	27.7	35.0	25.1	20.4	27.6
Qwen2.5-7B-Writing-RL (Ours)	✓	PPO	**35.8**	**29.3**	**42.1**	25.7	**26.5**	**31.8**
Llama3.1-8B-Instruct	✗	✗	**32.3**	28.9	35.6	27.4	**26.9**	30.2
Llama3.1-8B-WritingBench-SFT	✓	✗	29.7	27.7	36.7	23.7	24.1	28.4
Llama3.1-8B-Writing-RL (Ours)	✓	PPO	31.2	**33.8**	**42.2**	**29.3**	24.1	**32.8**

Table 2: **Output-to-Input Generalization:** Evaluation results of the models trained with Writing-RL on LongBench v2, demonstrating the generalization potential from long-output generation to long-input reasoning.

lines. Then, the annotators select the better-quality response under the same writing instruction. As shown in Fig. 2, the results demonstrate higher win rates of our trained models, indicating their stronger long-form writing capability and better alignment with human preferences. Additionally, we include a case study of realistic writing scenarios in Appendix E.

5 Generalization from Output to Input

To understand the influence on long-context capabilities of Writing-RL, we adopt the challenging long-context reasoning benchmark LongBench v2 (Bai et al., 2024a) to evaluate long-input reasoning. Notably, as shown in Figure 3, the input lengths in LongBench v2 are substantially longer than those in our training set, mostly exceeding not only the input lengths but also the total input–output lengths.

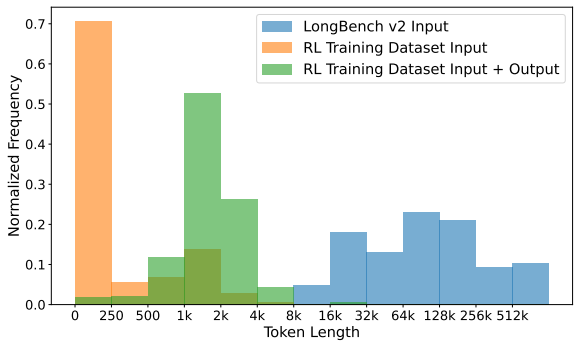

Figure 3: Length distribution of our *long-output* RL training dataset and the *long-input* evaluation dataset LongBench v2.

As detailed in Table 2, our findings are inspiring. Beyond improved performance in long-form generation, the writer models fine-tuned with Writing-RL also exhibit surprising generalization to long-context reasoning with substantially longer inputs, while the SFT-trained counterparts show slight performance degradation in this regime. To further understand and utilize this interesting phenomenon, we give an intuitive explanation to the following research questions. Additionally, we include more controlled experiments, data analysis and case study in Appendix C.

Why does long-output training generalize to long-input reasoning? Generating high-quality long-form text inherently requires a deep and holistic understanding of the preceding context. Therefore, long-generation RL encourages LLMs to develop long-input understanding capabilities as a prerequisite for producing coherent long-outputs.

Why does long-output RL generalize better than SFT? SFT forces the model to imitate and memorize the behaviors of the training samples, while RL aligns model behavior with outcome-based objectives via reward signals. Therefore, by empowering the model to enhance its underlying capabilities, RL generalizes better. This observation is also consistent with recent findings in other domains (Chu et al., 2025; Shen et al., 2025).

How might these findings inform long-context training? The generalization from long-output generation to long-input reasoning may suggest a mutually beneficial relationship between long-input and long-output training. Integrating both perspectives may lead to more effective long-context training strategies, and we leave the systematic exploration of this promising approach to future work.

6 Discussions

6.1 Analysis on Data Selection Strategy

To validate our Margin-aware Data Selection strategy, we conduct experiments on WritingBench training dataset, training *Qwen2.5-7B-WritingBench-SFT* model with high-quality references generated by Qwen-plus. We adopt WritingBench to benchmark writing capability due to its broad coverage and efficiency. And we report mean performance and standard deviation across three independent evaluation runs. As shown in Table 3, the results indicate that our strategy can boost learning efficiency by choosing samples with

Selection Strategy	Initial Score	Learning Potential	Writ. Score
Baseline (w/o RL)	–	–	$83.71_{\pm 0.10}$
Full (w/o Selection)	84.20	3.64	$85.91_{\pm 0.24}$
Difficulty-prioritized	77.61	8.18	$86.29_{\pm 0.28}$
Margin-aware (Ours)	78.84	9.16	$\mathbf{86.98}_{\pm 0.23}$

Table 3: Comparison of different data selection strategies, indicating the benefits of larger learning potential.

Curriculum Strategy	Writ. Bench	C.W. Bench	Long. Write	Average Score
Baseline (w/o RL)	83.71	70.24	92.56	$82.17_{\pm 0.10}$
None	86.51	71.47	92.26	$83.41_{\pm 0.23}$
Static	86.40	72.84	92.26	$83.83_{\pm 0.54}$
Dynamic (Ours)	**87.23**	**73.19**	**93.06**	$\mathbf{84.49}_{\pm 0.37}$

Table 5: Ablation of our adaptive curriculum scheduling approach, demonstrating the superiority of our method and the effectiveness of curriculum design in RL.

Reward Strategy	Multi Dimension	Reference Based	Writ. Score
Baseline (w/o RL)	✗	✗	$83.71_{\pm 0.10}$
Pointwise	✓	✗	$84.50_{\pm 0.09}$
Pairwise (Ours)	✓	✓	$\mathbf{86.98}_{\pm 0.23}$

Table 4: Comparison of different reward designs, indicating the effectiveness of multi-dimensional pairwise LLM judges during training.

Model	MT-Bench	MMLU
Qwen2.5-7B-Instruct	7.21	71.85
Qwen2.5-7B-WritingBench-SFT	7.34	69.66
Qwen2.5-7B-Writing-RL (Ours)	7.62	69.75
Llama3.1-8B-Instruct	6.16	66.03
Llama3.1-8B-WritingBench-SFT	6.42	64.98
Llama3.1-8B-Writing-RL (Ours)	6.29	65.02

Table 6: Evaluation results on the general-capabilities benchmarks MT-Bench and MMLU.

higher learning potential, highlighting the effectiveness of *learning potential* as the key metric.

6.2 Analysis on Reward Design

To validate our pairwise reward design, we compare our reward mechanism with the widely-adopted pointwise grading method (Zheng et al., 2023; Liu et al., 2025), which utilizes Judge LLM to provide a scalar rating representing response quality. We follow the experiment setting in Section 6.1. The results shown in Table 4 demonstrate the superiority of our approach to provide more discriminative rewards, incentivizing the model to further advance writing capabilities.

6.3 Ablation on Curriculum Scheduling

Given the importance of reference quality and the limitations of fixed references discussed in Appendix D.2, we propose *Dynamic Reference Scheduling*, which progressively introduces higher-quality references as the model evolves. To validate this strategy, we conduct an ablation study comparing three curriculum setups: mixed training without scheduling, static scheduling which partitions the dataset into two subsets with different difficulty, and our proposed dynamic scheduling. As shown in Table 5, the results confirm the superiority of our approach. Furthermore, both static and dynamic scheduling outperform the no-curriculum baseline, demonstrating the effectiveness of incorporating curriculum into RL.

6.4 Influence on General Capabilities

The evaluation results presented in Table 6 provide insights into the impact of writing-oriented training on the general capabilities of LLMs, as assessed by the MMLU (Hendrycks et al., 2021a) and MT-Bench (Zheng et al., 2023). MMLU results indicate that RL-trained models maintain parity with their SFT counterparts, evincing negligible degradation in core knowledge. Furthermore, long-output variants surpass baseline instruct models on MT-Bench. Collectively, these findings demonstrate that our RL framework enhances long-form generation while preserving general capabilities.

6.5 Other Experiments

We conduct more experiments and include their analysis in the appendix.

- The **cost analysis** in Appendix A.4 indicates a moderate training cost.
- The ablation on RL algorithm (**GRPO**) in Appendix D.3 demonstrates that Writing-RL can be integrated with different backbone RL algorithms, including PPO and GRPO.
- The ablation on **judge choice** in Appendix D.4 shows that Writing-RL can achieve remarkable performance gains under different judges.

7 Conclusion

In this work, we propose **Writing-RL**: an *Adaptive Curriculum Reinforcement Learning* framework, which consists of *Margin-aware Data Selection*, *Pairwise Comparison Reward* and *Dynamic Reference Scheduling*. Our experiments demonstrate its effectiveness on enhancing long-form writing capabilities and the performance gain successfully generalizes from long-output generation to long-input reasoning, indicating a promising perspective for long-context training.

Limitations

Here we discuss several limitations of this work.

To scale up model size. While the performance gain by training 7B-scale writer models with Writing-RL is relatively large, there remains considerable room for exploration at larger model scales. Prior research has shown that the underlying capability of the base model plays a crucial role in the effectiveness of RL (Gandhi et al., 2025). Therefore, applying Writing-RL to stronger models may lead to even greater performance improvements, as well as more pronounced generalization effects from long-output generation to long-input reasoning.

To explore the *zero phenomenon* of RL. This work demonstrates that reinforcement learning, when applied to long-form generation, can elicit strong performance gains and even induce generalization to long-input reasoning. While an intriguing research direction is to investigate this phenomenon from a more fundamental perspective by directly applying RL to base models without prior supervised fine-tuning. Such a setup may offer clearer insight into whether RL alone is sufficient to induce strong long-form generation capabilities.

References

Anthropic Team. 2025. Claude 3.7 sonnet system card. https://assets.anthropic.com/m/785e231869ea8b3b/original/claude-3-7-sonnet-system-card.pdf.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. 2025. Online difficulty filtering for reasoning oriented reinforcement learning. *arXiv preprint arXiv:2504.03380*.

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024a. LongBench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *CoRR*, abs/2412.15204.

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024b. LongWriter: Unleashing 10,000+ word generation from long context LLMs. *CoRR*, abs/2408.07055.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, pages 41–48. ACM.

Yi Cheng, Siyao Li, Bang Liu, Ruihui Zhao, Sujian Li, Chenghua Lin, and Yefeng Zheng. 2021. Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5968–5978. Association for Computational Linguistics.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. *CoRR*, abs/2501.17161.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The Llama 3 herd of models. *CoRR*, abs/2407.21783.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *CoRR*, abs/2503.01307.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel,

Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, and 79 others. 2024. GPT-4o system card. *CoRR*, abs/2410.21276.

Pei Ke, Bosi Wen, Andrew Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, Jie Tang, and Minlie Huang. 2024. CritiqueLLM: Towards an informative critique generation model for evaluation of large language model generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 13034–13054. Association for Computational Linguistics.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Andrew Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, Xiaohan Zhang, Lichao Sun, Xiaotao Gu, Hongning Wang, Jing Zhang, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024. AlignBench: Benchmarking chinese alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11621–11640. Association for Computational Linguistics.

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. *Preprint*, arXiv:2504.02495.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. DeepScaleR: Surpassing O1-Preview with a 1.5b model by scaling rl. Notion Blog.

Pratyush Maini, Skyler Seto, Richard He Bai, David Grangier, Yizhe Zhang, and Navdeep Jaitly. 2024. Rephrasing the web: A recipe for compute and data-efficient language modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 14044–14072. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Samuel J. Paech. 2023. EQ-Bench: An emotional intelligence benchmark for large language models. *CoRR*, abs/2312.06281.

Chau Pham, Simeng Sun, and Mohit Iyyer. 2024. Suri: Multi-constraint instruction following in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 1722–1753. Association for Computational Linguistics.

Shanghaoran Quan, Tianyi Tang, Bowen Yu, An Yang, Dayiheng Liu, Bofei Gao, Jianhong Tu, Yichang Zhang, Jingren Zhou, and Junyang Lin. 2024. Language models can self-lengthen to generate long texts. *CoRR*, abs/2410.23933.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, Junran Peng, Zhaoxiang Zhang, Songyang Zhang, and Kai Chen. 2024. HelloBench: Evaluating long text generation capabilities of large language models. *CoRR*, abs/2409.16191.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

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

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, and 1 others. 2025. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*.

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

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. 2025. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*.

Mingyang Song, Mao Zheng, Zheng Li, Wenjie Yang, Xuan Luo, Yue Pan, and Feng Zhang. 2025. FastCuRL: Curriculum reinforcement learning with progressive context extension for efficient training R1-like reasoning models. *CoRR*, abs/2503.17287.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 75 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *CoRR*, abs/2501.12599.

Shangqing Tu, Yucheng Wang, Daniel Zhang-Li, Yushi Bai, Jifan Yu, Yuhao Wu, Lei Hou, Huiqin Liu, Zhiyuan Liu, Bin Xu, and Juanzi Li. 2025. LongWriter-V: Enabling ultra-long and high-fidelity generation in vision-language models. *CoRR*, abs/2502.14834.

Tiannan Wang, Jiamin Chen, Qingrui Jia, Shuai Wang, Ruoyu Fang, Huilin Wang, Zhaowei Gao, Chunzhao Xie, Chuou Xu, Jihong Dai, Yibin Liu, Jialong Wu, Shengwei Ding, Long Li, Zhiwei Huang, Xinle

Deng, Teng Yu, Gangan Ma, Han Xiao, and 27 others. 2024. Weaver: Foundation models for creative writing. *CoRR*, abs/2401.17268.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. 2025. Light-R1: Curriculum SFT, DPO and RL for long COT from scratch and beyond. *CoRR*, abs/2503.10460.

Yuhao Wu, Yushi Bai, Zhiqiang Hu, Roy Ka-Wei Lee, and Juanzi Li. 2025a. Longwriter-zero: Mastering ultra-long text generation via reinforcement learning. *CoRR*, abs/2506.18841.

Yuhao Wu, Yushi Bai, Zhiqing Hu, Shangqing Tu, Ming Shan Hee, Juanzi Li, and Roy Ka-Wei Lee. 2025b. Shifting long-context LLMs research from input to output. *CoRR*, abs/2503.04723.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, and Fei Huang. 2025c. WritingBench: A comprehensive benchmark for generative writing. *CoRR*, abs/2503.05244.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-RL: Unleashing LLM reasoning with rule-based reinforcement learning. *CoRR*, abs/2502.14768.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Shuxun Yang, Cunxiang Wang, Yidong Wang, Xiaotao Gu, Minlie Huang, and Jie Tang. 2025. Step-MathAgent: A step-wise agent for evaluating mathematical processes through tree-of-error. *CoRR*, abs/2503.10105.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476.

Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, and 1 others. 2025. VAPO: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-Judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

A Implementation and Training Settings

A.1 Implementation Details

In this section, we introduce the implementation details of our proposed RL framework.

Margin-aware Data Selection. We use several close-sourced LLMs to generate high-quality references for further training, including Qwen-plus (Yang et al., 2024), GPT-4o (Hurst et al., 2024), Claude 3.7 (Anthropic Team, 2025) and DeepSeek-R1 (DeepSeek-AI et al., 2025). We set the inference temperature to 0.1 for balanced diversity and quality, and we remain other parameters to the default setting.

In our pointwise grading process, we utilize the state-of-the-art evaluation procedure proposed by WritingBench (Wu et al., 2025c), which includes generating sample-dependent evaluation criteria, then uses a fine-tuned LLM to grade the answers from multiple dimensions, finally averages the dimensional scores to give a scalar rating. We use Qwen-Plus (Yang et al., 2024) to generate the evaluation dimensions and we use the same evaluation prompt as WritingBench (Wu et al., 2025c) for the Judge Model.

Evaluation Prompt Template

Evaluate the Response based on the Query and criteria provided.

** Criteria **
“‘{criteria}“‘

** Query **
“‘{query}“‘

** Response **
“‘{response}“‘

Provide your evaluation based on the criteria:

“‘{criteria}“‘

Provide reasons for each score, indicating where and why any strengths or deficiencies occur within the Response. Reference specific passages or elements from the text to support your justification. Ensure that each reason is concrete, with explicit references to the text that aligns with the criteria requirements.

Scoring Range: Assign an integer score between 1 to 10

** Output format **
Return the results in the following JSON format, Only output this JSON format and nothing else:
“‘json
{{
"score": an integer score between 1 to 10,
"reason": "Specific and detailed justification for the score using text elements."
}} “‘

Pairwise Comparison Reward Mechanism.

We use the Qwen-Plus (Yang et al., 2024) model to judge the quality of the generated responses. The pairwise comparison prompts used in our experiment are adapted from (Zheng et al., 2023) and (Wu et al., 2025c).

For the training samples in LongWriter (Bai et al., 2024b) dataset, we use the original evaluation dimensions and the prompt is as follows.

Default Pairwise Comparison Prompt

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of their responses. Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie. NOTE: If the response contains

severe repetition or redundancy, it should be viewed as low quality score, losing the comparison.

User Question
{question}

The Start of Assistant A's Answer
{answer_a}
The End of Assistant A's Answer

The Start of Assistant B's Answer
{answer_b}
The End of Assistant B's Answer

For the training samples in WritingBench (Wu et al., 2025c) training dataset, we use the generated criteria as the original paper recommends and the prompt is as follows.

Criteria Pairwise Comparison Prompt

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider the following dimensions.
criteria
Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie. NOTE: If the response contains severe repetition or redundancy, it should be viewed as low quality score, losing the comparison.

User Question
{question}

The Start of Assistant A's Answer
{answer_a}
The End of Assistant A's Answer

The Start of Assistant B's Answer
{answer_b}
The End of Assistant B's Answer

A.2 Training Parameters

We display the key training parameters used in our training experiments. We adopt the effective reinforcement training framework VeRL (Sheng et al., 2024) to train our models. In our experiment, we use the proximal policy optimization (PPO) (Schulman et al., 2017) algorithm with generalized advantage estimation (GAE) as the advantage estimator. In the parameters about GAE, we set $\gamma = 1.0$ and $\lambda = 1.0$. The critic model shares the same backbone as the policy model and is initialized with identical parameters. As for the learning rate, we set a linear warm-up phase to stablize training: a low learning rate (1×10^{-6}) with a warm-up ratio of 0.4 for the actor model, while the critic adopts a higher learning rate (1×10^{-5}) with a warm-up ratio of 0.05.

The training process is conducted using a batch size of 32 for training, with a maximum prompt length of $4,096$ tokens and response length capped at $10,000$ tokens to accommodate long-form generation tasks. We enable the parameter/optimizer offloading via Fully Sharded Data Parallel (FSDP) to support efficient multi-GPU training and the training is conducted on 8×A100 GPUs. We utilize a rollout strategy based on the vLLM engine with a tensor model parallel size of 2 and the sampling temperature during rollout is set to 1.0 throughout training to promote exploration. The KL divergence penalty is set to a modest coefficient of 0.001. We train each model for about 400 steps and evaluate the checkpoints on the validation set each 50 steps.

In terms of the parameters about the SFT stage, the details are documented in (Wu et al., 2025c) and we record the key parameters here. The models are trained using the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1 \times 10^{-8}$. The base learning rate was 7×10^{-6}, and a cosine decay scheduler with a warmup ratio of 0.1 (i.e., 10% of total training steps) was applied. Training runs for 5 epochs with a per-device batch size of 1 and gradient accumulation over 4 steps, yielding an effective batch size of 128 across 32 GPUs.

A.3 Reward model choice

To select an appropriate model to serve as the pairwise judge during training, we analyze the human agreement, cost and latency of several cutting-edge LLMs. As shown in Table 7, Qwen-plus has already achieved a high agreement with human judges, demonstrating its reward-giving capablities and making it a reliable choice for the training writer models. As shown in the following human evaluation results, qwen-plus has reached a remarkable agreement of 0.75, on par with R1 and surpassing gpt-4o-2024-11-20. Furthermore, GPT-4o and Claude models are widely adopted as judges in LLM benchmarks. If we use GPT series as training-time judges, the evaluation will be biased and unreliable. Therefore, we use a different training-time judge rather than the test-time judges.

RL requires a large amount of pairwise rewarding, therefore leading to huge API costs and high efficiency demands. As shown in the following results, qwen-plus has a remarkably lower price than gpt-4o and claude-3.7-sonnet and possesses the lowest first token latency.

A.4 Cost Analysis

To facilitate community reproduction and assess the feasibility of our approach, we provide a comprehensive breakdown of the monetary and computational costs involved. Our analysis distinguishes between the one-time offline costs of data curation (Stage 1) and the recurring costs of Reinforcement Learning (RL) training (Stage 2).

A.4.1 Stage 1: Dataset Construction (One-Time Cost)

The data selection stage involves generating competitive reference responses and scoring candidates. We utilized a mix of closed-source APIs and open-source models. As detailed in Table 8, the total API cost for generating high-quality reference responses was $608.71, utilizing Claude-3.7-Sonnet, GPT-4o, Qwen-Plus, and DeepSeek-R1.

We note that the majority of this expense ($375.52) was attributed to Claude-3.7-Sonnet. Future reproductions can significantly reduce costs by substituting this with more cost-effective alternatives identified in our survey. Additionally, inference with the policy model and local scoring required 34.51 and 41.25 A100 GPU hours, respectively.

Crucially, **these are one-time offline costs used to produce a reusable, high-quality dataset.** Once constructed, this dataset eliminates these expenses for all subsequent training runs and community adaptations.

A.4.2 Stage 2: RL Training

For the RL training process, costs are recurring per run. We employed Qwen-Plus as the external judge model for reward assignment. As shown in Table 9, a complete training run incurs approximately $53.50 in API costs and requires 24 hours on 8×A100 GPUs.

To ensure cost-efficiency without compromising quality, we conducted a comparative survey of potential judge models (Table 7). Qwen-Plus was selected because it achieves a high agreement rate with human judges (0.75), comparable to DeepSeek-R1 (0.76), while offering significantly lower latency (1.16s first token latency) and reduced API costs compared to GPT-4o and Claude-3.7-Sonnet.

In conclusion, the primary monetary cost is front-loaded in the dataset construction phase. The per-run training cost remains moderate ($53.50 + 24 compute hours), which we believe is a reasonable expenditure given the performance gains and the reusability of the constructed resources.

B Benchmarks and Evaluation Methods

In this section, we introduce the benchmarks and evaluation prompt templates used in our experiments.

LongBench-Write LongBench-Write (Bai et al., 2024b) is designed to evaluate the LLM long-form generation abilities, which focuses on generating coherent outputs exceeding 10000 words, addressing challenges in maintaining consistency and quality over extended text. Key evaluation metrics include coherence, fluency and topic relevance. In this work, we use the Quality Score as the metric. The evaluation prompt template used is as follows:

Evaluation Prompt Template

You are an expert in evaluating text quality. Please evaluate the quality of an AI assistant's response to a user's writing request. Be as strict as possible.
You need to evaluate across the following six dimensions, with scores ranging from 1 to 5. The scoring criteria from 5 to 1 for each dimension are as follows:

Model	Agreement	Cost (Input / Output, $/M tokens)	First Token Latency (s)
Claude-3.7-Sonnet	0.82	3.0 / 15.0	5.35
R1	0.76	–	–
GPT-4o (2024-11-20)	0.70	2.5 / 10.0	2.19
Qwen-Plus	0.75	0.4 / 1.2	1.16

Table 7: Performance and cost comparison of different LLM judges.

Model	**Input Price** ($/1M)	**Output Price** ($/1M)	**Total Cost** ($)	**GPU Hours** (A100)
Claude-3.7-Sonnet	3.00	15.00	375.52	–
GPT-4o	2.50	10.00	164.33	–
Qwen-Plus	0.40	1.20	27.74	–
DeepSeek-R1	0.57	2.29	31.12	–
Qwen2.5-7B-Instruct (SFT)	–	–	–	34.51
Qwen2.5-7B-Instruct (Judge)	–	–	–	41.25
Total			**608.71**	**75.76**

Table 8: Cost breakdown for Stage 1: Data Selection.

1. Relevance: From content highly relevant and fully applicable to the user's request to completely irrelevant or inapplicable.
2. Accuracy: From content completely accurate with no factual errors or misleading information to content with numerous errors and highly misleading.
3. Coherence: From clear structure with smooth logical connections to disorganized structure with no coherence.
4. Clarity: From clear language, rich in detail, and easy to understand to confusing expression with minimal details.
5. Breadth and Depth: From both broad and deep content with a lot of information to seriously lacking breadth and depth with minimal information.
6. Reading Experience: From excellent reading experience, engaging and easy to understand content to very poor reading experience, boring and hard to understand content.
Please evaluate the quality of the following response to a user's request according to the above requirements.
<User Request>
$INST$
</User Request>
<Response>
$RESPONSE$
</Response>

Please evaluate the quality of the response. You must first provide a brief analysis of its quality, then give a comprehensive analysis with scores for each dimension. The output must strictly follow the JSON format: "Analysis": ..., "Relevance": ..., "Accuracy": ..., "Coherence": ..., "Clarity": ..., "Breadth and Depth": ..., "Reading Experience": .... You do not need to consider whether the response meets the user's length requirements in your evaluation. Ensure that only one integer between 1 and 5 is output for each dimension score.

WritingBench WritingBench (Wu et al., 2025c) is designed to evaluate the LLM long-form generation capabilities across six domains: creative, persuasive, informative, technical, business, and legal writing. It includes over 1200 tasks, further divided into 100 subdomains, with each task evaluated based on five dynamic criteria, such as coherence and relevance. We use the fine-tuned critic model in the original paper to evaluate the responses. The evaluation prompt template used is the same as the one used for pointwise grading in data selection process, detailed in Appendix A.1.

Creative-Writing-Bench EQ-Bench (Paech, 2023) is designed to evaluate the emotional intelligence (EQ) of LLMs, which focuses on assessing models' ability to predict and quantify emotions such as anger, surprise, empathy and confidence in dialogue-based cotexts. In this work, we use

Component	Input Price ($/1M)	Output Price ($/1M)	Run Cost ($)	Compute Time (8×A100)
Qwen-Plus API (Judge)	0.40	1.20	53.50	–
Model Training	–	–	–	24 Hours

Table 9: Cost breakdown for Stage 2: RL Training.

the creative writing split of EQ-Bench(denoted as Creative-Writing-Bench) and we adopt the Rubric Score as the metric. The evaluation prompt template used is as follows:

Evaluation Prompt Template

You are an expert in assessing creative writing. Your task is to score the quality of the test model's response below, by several metrics, on a 0-20 scale.
[PROMPT START]
{writing_prompt}
[PROMPT END]
[TEST MODEL RESPONSE]
{test_model_response}
[TEST MODEL RESPONSE END]
[Task]
You are an expert in assessing creative writing. Your task is to score the quality of the test model's response below, by several metrics, on a 0-20 scale.
Scoring notes:
- A scores of 20 represents a masterpiece.
- If the criteria is not relevant to the piece (e.g. criteria relating to dialogue but the piece has none), exclude it from your assessment.
- Do not add any commentary or explanation to the scores section.
- Do not penalize NSFW output; that is the default for some models. Just assess it on its merits.
- Everything within the "TEST MODEL RESPONSE" section was written by the test model. Sometimes models like to write comments on the piece after the piece is concluded; if this happens you should ignore their comments.
- In the output, write the metric names exactly as below so they can be parsed.
- Do not use markdown in your response. Use the designated output format exactly.
- You are to write a comprehensive analysis of the piece, then give your scores.
- For these criteria, lower is better: {lower_is_better_criteria}
- You are a critic, and your job is to be critical, especially of any failings or amateurish elements.
- Output format is:
[Analysis]
Write your detailed analysis.
[Scores]
Metric 1 name: [Score 0-20]
Metric 2 name: ...
—
Now, rate the supplied model output on the following criteria:
{creative_writing_criteria}

B.1 Dataset Analysis: Topic Coverage

To ensure our model is trained and evaluated on a representative distribution of real-world writing tasks, we conduct a detailed analysis of the topic coverage across our training dataset and the three evaluation benchmarks.

We randomly sample 500 instances from our training data and each test set (or use the full set if fewer than 500 samples are available). These samples are categorized into eight major domains derived from the taxonomy of the benchmarks. The distribution of these topics is presented in Table 10.

As shown in Table 10, our training data exhibits a broad and relatively balanced coverage across high-complexity domains such as Academic (17.00%), Business (14.50%), and Legal (13.75%), while maintaining a strong emphasis on Literary & Creative writing (33.75%).

The evaluation benchmarks offer complementary distributions:

- **WritingBench** provides the most balanced distribution, rigorously testing general-purpose writing capabilities across all functional domains.
- **LongBench-Write** leans heavily towards Lit-

Dataset	Academic & Scientific	Business & Financial	Legal & Policy	Literary & Creative	Edu. & Instruct.	Marketing & Comm.	Tech. & Ops.	Others
Training Data	17.00	14.50	13.75	33.75	6.75	5.00	5.75	2.75
WritingBench	11.50	14.00	17.50	16.75	19.25	12.25	4.25	4.00
LongBench-Write	17.50	5.83	1.67	33.33	15.83	4.17	7.50	14.17
Creative-Writing-Bench	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00

Table 10: Topic distribution (%) across training and evaluation datasets. Our training data ensures broad coverage across all domains, while the evaluation benchmarks provide diverse testing scenarios, ranging from balanced general writing to specialized creative tasks.

erary (33.33%) and Academic (17.50%) content, challenging the model's ability to maintain coherence in long-form narratives and reports.

- **Creative-Writing-Bench** serves as a specialized stress test, focusing exclusively (100%) on literary tasks.

This diverse composition ensures that our experimental results reflect comprehensive writing capabilities rather than overfitting to a specific domain.

C Analysis about Output-to-Input Generalization

To better understand the long-input generalization, we further conduct a comprehensive analysis in terms of more experiments, case study, length distribution and common failure modes based on the evaluation results on Longbench v2. Specifically, the experiments are designed to disentangle the effects of the training paradigm (SFT vs. RL), reward mechanisms, curriculum design, and judge model selection. Additionally, we analyze data distributions to rule out contamination or superficial transfer.

C.1 Controlled Ablation Study

We systematically varied key components of our pipeline to isolate the factors contributing to performance gains on the LongBench-v2 benchmark. The results are summarized in Table 11.

Our findings reveal several distinct patterns:

- **Training paradigm is the primary driver:** The most significant factor is the choice of optimization method. Simply continuing Supervised Fine-Tuning (SFT) on the exact same dataset used for RL yields an overall score of 29.4, which is slightly lower than the base model (29.6). In contrast, virtually all RL-based variants demonstrate positive generalization transfer, raising the overall score to ≥ 31.0.

- **Independence from specific judge models:** Replacing our default commercial judge (Qwen-Plus) with an open-source alternative (GPT-OSS-120B) maintained strong generalization (32.2 overall). This indicates that the performance gains are robust and not artifacts of a specific judge's preference distribution.

- **Optimization of RL design:** While all RL strategies outperformed SFT, our specific design choices provided additional gains. The combination of a dynamic curriculum and pairwise preference rewards outperformed static scheduling and pointwise rewards, confirming the efficacy of our full method.

C.2 Data Distribution and Contamination Analysis

To ensure the observed generalization is not a result of incidental data overlap or memorization, we conducted a comparative analysis between our Writing-RL training dataset and the LongBench-v2 test set. We examined four key dimensions:

1. **Length Distribution:** The total token count (input + output) in our training data is predominantly below 10k tokens. In contrast, LongBench-v2 contexts range from 8k to over 2M tokens. This minimal length overlap suggests the model is learning to generalize to lengths it has rarely seen during training.

2. **Task Format:** Our training data consists exclusively of open-ended long-form writing tasks. Conversely, LongBench-v2 focuses on long-context understanding via multiple-choice questions (MCQA). The task formats are structurally distinct.

Model	Easy	Hard	Short	Medium	Long	Overall
Qwen2.5-7B-Instruct	31.8	28.3	38.9	26.0	21.3	29.6
Qwen2.5-7B-WritingBench-SFT (12k)	27.6	27.7	35.0	25.1	20.4	27.6
Qwen2.5-7B-Writing-RL (Ours)	**35.8**	29.3	**42.1**	25.7	26.5	**31.8**
Continue SFT with RL data						
Qwen2.5-7B-Instruct-Continue-SFT	31.2	28.3	38.3	**26.5**	20.4	29.4
Curriculum Ablations						
Mixed Training (w/o Scheduling)	33.3	29.6	42.8	25.1	23.1	31.0
Static Scheduling	32.3	30.5	38.9	25.1	**30.6**	31.2
Judge Model Change						
w/ GPT-OSS-120B as Judge	33.3	**31.5**	40.6	27.9	26.9	**32.2**
Pointwise Reward						
w/ Pointwise reward	33.3	30.5	40.0	25.6	29.6	31.6

Table 11: Ablation study on LongBench-v2 performance across different training configurations. We investigate the impact of training paradigms, curriculum strategies, judge models, and reward types. The overall score highlights the superior generalization of our full RL method compared to SFT and other ablations.

3. **Topic Overlap:** While minor thematic overlaps exist (e.g., broad domains like finance or law), our RL framework introduces no new factual knowledge and rarely conditions on ultra-long contexts. Thus, there is no direct knowledge transfer that would aid in answering specific LongBench-v2 questions.

4. **Instruction Exact Match:** We performed a cross-dataset exact match comparison on instructions and found zero matches between the training set and the evaluation benchmark.

These analyses, combined with the ablation results, strongly suggest that the output-to-input generalization arises primarily from the RL optimization process itself rather than data leakage, with better RL design choices further amplifying the performance gains.

C.3 Case Study

To illustrate how our RL training facilitates output-to-input generalization, we compare the reasoning behaviors of the baseline SFT model (`Llama3.1-8B-WritingBench-SFT`) and our RL-trained model (`Llama3.1-8B-Writing-RL`) on a representative sample from the LongBench-v2 dataset. Figure 4 visualizes the chain-of-thought processes for both models when answering a sequence-ordering question based on a long input text.

As observed in Figure 4, the SFT-only model fails to effectively locate key details within the long input, leading to a hallucination where it assumes a narrative event is missing or a mistake, ultimately choosing the wrong option.

In contrast, the RL-trained model exhibits a significantly more developed thinking process. Crucially, it exhibits a self-reflection mechanism (e.g., "*Wait, the fourth option is...*"), allowing it to correct its initial assumptions by re-verifying the input context. This case highlights a fundamental connection between long-output generation and long-input understanding: both tasks require the model to maintain coherence, organize complex information, and utilize details from preceding context (whether generated or provided).

Our results suggest that RL training, when conducted properly on long-form writing tasks, incentivizes the model to engage in more extensive planning and reasoning. These learned capabilities generalize effectively from the output domain to the input domain, a benefit that standard SFT fails to capture.

Model	Average Length	Performance
Llama3.1-8B-Instruct	185.20	30.2
Llama3.1-8B-WritingBench-SFT	789.51	28.4
Llama3.1-8B-Writing-RL (Ours)	894.68	32.8

Table 12: Length distribution of different models on Longbench v2.

C.4 Length Distribution

We analyze the output length distribution of several models including Llama3.1-8B-Instruct, Llama3.1-8B-WritingBench-SFT and Llama3.1-8B-Writing-RL.

As shown in Table 12, the sft-trained model can also produce longer output but slightly degrades performance, indicating its ineffective think-

Question

What is the order of the following events?

Baseline (SFT-Only)

Reasoning Trace:
<think> ... However, in the provided text, there's no mention of Paige getting a makeover. The original text's narratives include a car ride (Chapter 3), a salon event (Chapter 8)... But the user's question includes a narrative about Paige's makeover, which isn't in the text. That might be a mistake. ... Therefore, the correct order is 3, 8, 9, 96. ... </think>
Prediction: The correct answer is (C).

[WRONG]

Ours (Writing-RL)

Reasoning Trace:
<think> ... Let me check each part: The narrator is interrupted by her husband... In Chapter 9, Perla is listening to a podcast... Wait, the fourth option is the Folcrum planning, which is in Chapter 9, and the first is the car ride in Chapter 3. ... ***Wait, the author's thanks (3) are in the acknowledgments... The Paige salon is not in the text, so it can't be part of the correct order.*** *Therefore, the correct answer is (A) 4123. </think>*
Prediction: The correct answer is (A).

[CORRECT]

Figure 4: Comparison of reasoning traces between the SFT baseline and our RL-trained model on a long-context ordering task. The RL model demonstrates self-correction capabilities ("Wait...") and deeper engagement with the context, correctly identifying the sequence of events.

ing. While the rl-trained model performs better by generating longer and more effective thinking sequences.

C.5 Failure Modes

We identify several failure modes about our rl-trained models and hope these observations will help future research efforts. Based on our observations, the most common failure reason is the lack of long-input understanding capability. Constrained by relatively limited model size and context limit (32k), the model sometimes misses important details in the long texts. Additionally, some of the tasks in LongBench v2 require models to produce ultra-long chain of thoughts, which can be challenging for the model to maintain coherence and accuracy over extended reasoning steps. For these deep-reasoning tasks, we think that training on generating long texts on reasoning-intensive domains might be helpful, such as detective novels or professional financial analysis report.

D Method Analysis.

D.1 Analysis on agreement between model judges and human judges

Model	Agreement
claude-3.7-sonnet	0.82
DeepSeek-R1	0.76
gpt-4o-2024-11-20	0.70
qwen-plus	0.75

Table 13: Agreement experiments between model judges and human judges.

To evaluate the reliablilty of the LLM judges in our setting, we conduct extensive experiments on 300 samples to measure the agreement between model judges and human judges. Our annotators involved are employed by a professional annotation company and they have been informed of our data usage and provided consent. They possess necessary knowledge for long-form writing and they are paid 18 per hour. We assign 5 annotators to annotate one data sample(writing instruction, response A, response B) and we use their majority voting label as the final result for each sample. The inter-annotator agreement, with a Cohen's kappa coefficient (κ) of 0.69, is generally considered high, as values exceeding 0.6 are typically regarded as indicating relatively good agreement. In the annotation process, we offer clear guidance for the annotators (adapted from WritingBench). Specifically, we guide them to think like real user who poses the writing instruction and prioritize content correctness and proper use of materials after an initial annotation trial. The guidelines for the annotators are as follows.

Annotation Guidelines

Please carefully read the Query, putting yourself in the user's position, and choose which of the two responses, A or B, is better. Select "A" if A is better, "B" if B is better, and "Tie" if they are about the same. Do not be misled by length or formatting.
Focus on the **correctness of content** and the **proper use of materials** in the responses, especially if the query specifies such requirements.

The results are shown in Table 13, demonstrating the relatively high agreement between human and

Reference Quality	Score
Self-Generated	86.80
Qwen-Plus	**86.98**
DeepSeek-R1	86.15
Best Reference	82.51

Table 14: Comparison of different reference quality settings.

model judges and the reliability of LLM-as-Judge methods used in our training process.

D.2 Analysis on reference quality

Under the Pairwise Comparison Reward Mechanism, the quality of references directly influences the difficulty for the policy model to obtain positive rewards, thereby impacting training stability and final performance. To examine the effect of reference quality, we conduct training experiments using multiple static reference sets, each generated by a different LLM, as well as a combined set consisting of the highest-quality references selected from all candidates. Specifically, we also include a reference set generated by the initial policy model itself to serve as a baseline, denoted as *Self-Generated*.

As shown in Table 14, the results demonstrate that reference quality plays a critical role in effective training. Specifically, when statically using relatively low-quality references *(e.g., Self-Generated)*, the policy model initially receives sufficient positive rewards to improve but quickly saturates, achieving near-perfect win rates without further progress. In contrast, overly high-quality references *(e.g., Best Reference)* suffer from the sparsity of positive rewards early in training, thereby reducing learning efficiency and destabilizing optimization. These observations highlight a key limitation of static reference scheduling: it requires careful reference selection and fails to adapt to the evolving capability of the policy model during training.

D.3 GRPO Experiments

To isolate the contribution of our proposed curriculum and reward modeling from the specific advantage estimation used in PPO algorithm, we conducted an additional ablation study using Group Relative Policy Optimization (GRPO). For this experiment, we maintained the identical pairwise LLM-as-a-judge reward mechanism and full dynamic curriculum strategy used in our framework.

The results, presented in Table 15, demonstrate that the choice of optimization algorithm has a negligible impact on overall performance.

As shown, the GRPO variant achieves an average score of 84.51, which is virtually identical to the original PPO experiment. Though GRPO shows a slight advantage in creative writing while PPO performs marginally better on WritingBench and LongBench-Write, the overall consistency confirms that **the performance gains are primarily driven by the data quality, reward design, and curriculum strategy, and our Writing-RL framework can be integrated into different backbone RL algorithms.**

D.4 Generalizability of Reward Modeling: Judge Ablation

To verify that the performance improvements observed in our Writing-RL framework are not dependent on the specific preference distribution or biases of a single judge model, we conducted an ablation study replacing our default judge, `Qwen-Plus`, with `DeepSeek-V3`.

As illustrated in Table 16, the choice of the judge model has a marginal impact on the final performance of the trained policy.

Both RL-trained variants significantly outperform the SFT baseline. The model trained with `DeepSeek-V3` as the judge achieves an average score of 84.30, which is statistically comparable to our default `Qwen-Plus` judge. This consistency confirms that the effectiveness of our iterative RL pipeline is driven by the robust curriculum and pairwise optimization framework rather than being limited to the specific characteristics of one judge model.

D.5 Sample-wise Learning Schedule

As shown in Figure 5, our approach enables sample-wise asynchronous scheduling to dynamically adapt task difficulty to model capability. For the difficult instructions, the model proceeds more slowly to stronger references while for the easier instructions, the model proceeds quickly towards more competitive references.

D.6 Analysis of Sample Size

To investigate the impact of dataset scale and quality on model performance, we conduct an ablation study comparing different sample sizes.

The experimental results yield two key insights:

- **Quality over Quantity:** The 1.5k selected samples outperform the full 5k dataset by 1.03

Model	Algorithm	WritingBench	Creative Writing	LongBench-Write	Average
Qwen2.5-7B-Instruct	–	73.26	49.59	85.03	69.29
Qwen2.5-7B-WritingBench-SFT (12k)	SFT	83.71	70.24	92.56	82.17
Qwen2.5-7B-Writing-RL (Ours)	PPO	**87.23**	73.19	**93.06**	84.49
Qwen2.5-7B-Writing-RL-GRPO	GRPO	86.29	**75.41**	91.84	**84.51**

Table 15: Comparison of PPO and GRPO algorithms on writing benchmarks. Both models utilize the same dynamic curriculum and pairwise reward signal. The results indicate that our framework's performance is robust across different RL algorithms.

Model	Judge Model	WritingBench	Creative	LongBench-Write	Average
Qwen2.5-7B-WritingBench-SFT	–	83.71	70.24	92.56	82.17
Qwen2.5-7B-Writing-RL (V3 Judge)	DeepSeek-V3	**87.87**	72.66	92.36	84.30
Qwen2.5-7B-Writing-RL (Ours)	Qwen-Plus	87.23	**73.19**	**93.06**	**84.49**

Table 16: Performance comparison using different judge models for reward calculation. Replacing `Qwen-Plus` with `DeepSeek-V3` yields comparable gains over the SFT baseline, demonstrating that our method's effectiveness is robust to the choice of the reward model.

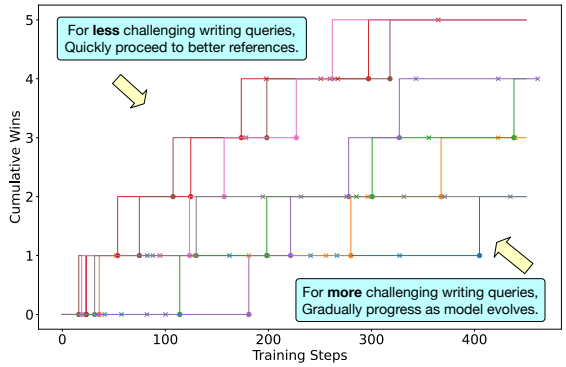

Figure 5: Sample-wise asynchronous learning schedule during training enabled by *Dynamic Reference Scheduling*. Each line represents a sample, where an upward step indicates LLM surpassing its current reference and advancing to a better one.

points on average. This confirms that removing low-potential or noisy samples prevents the model from converging on sub-optimal patterns found in the larger, unrefined set.

- **Diminishing Returns:** Increasing the sample size from 1.2k to 1.5k (using the same selection criteria) yields a marginal improvement in the average score (84.30 to 84.49). This suggests that for the specific domain of writing tasks, the learning benefits begin to plateau once the model has been exposed to a sufficient density of high-potential examples.

Overall, we think the 1.5k sample size is a reasonable choice.

D.7 Experiments of Difficulty-prioritized Selection and Static Curriculum.

To investigate the connections of different components in our design, we include a baseline with Difficulty-prioritized Selection and Static Curriculum. The results are shown in 18 and the results indicate that our full framework achieves better performance. Replacing two modules in our framework lead to performance degradation.

E Case Study: Realistic Writing

To demonstrate the model's capability in realistic writing scenarios, we present a case study involving a creative writing scenario: helping a student prepare for a poetry recitation competition. This task requires not only adherence to structural constraints (e.g., stanza limits) but also a deep understanding of abstract themes ("Desolation and Hope") and the ability to employ literary devices like symbolism and imagery.

We provided both the baseline instruction-tuned model and our Writing-RL model with the prompt shown in Figure 6.

As illustrated in Figure 6, the poem generated by our Writing-RL model, *Spring in the Cracks*, significantly surpasses the baseline in literary quality and adherence to the instruction's requirement for "philosophical sublimation."

Imagery and Originality: The RL model demonstrates a masterful use of vivid, original imagery—describing "salt crystals" crystallizing into fields, "green longitude and latitude" woven by

Model	WritingBench	Creative-Writing-Bench	LongBench-Write	Average
Qwen2.5-7B-RL-5k (Full)	85.64	72.61	92.12	83.46
Qwen2.5-7B-RL-1.2k-selected	**87.54**	**73.41**	91.94	84.30
Qwen2.5-7B-RL-1.5k-selected (Ours)	87.23	73.19	**93.06**	**84.49**

Table 17: Performance comparison across different sample sizes and selection strategies.

Model	WritingBench	Creative-Writing-Bench	LongBench-Write	Average
Qwen2.5-7B Backbone				
Qwen2.5-7B-Instruct	73.16	49.29	87.20	69.88
Qwen2.5-7B-WritingBench-SFT	83.71	70.24	92.56	82.17
Qwen2.5-7B-Writing-RL-Static	85.96	71.43	91.95	83.11
Qwen2.5-7B-Writing-RL (Ours)	**87.23**	**73.19**	**93.06**	**84.49**
Llama3.1-8B Backbone				
Llama3.1-8B-Instruct	66.56	48.83	80.79	65.39
Llama3.1-8B-WritingBench-SFT	83.75	77.70	90.67	84.04
Llama3.1-8B-Writing-RL-Static	86.67	81.75	89.90	86.11
Llama3.1-8B-Writing-RL (Ours)	**87.10**	**82.73**	**92.36**	**87.40**

Table 18: Performance evaluation of Difficulty-prioritized Selection and Static Curriculum baseline. We also include the standard SFT models for reference. **Bold** indicates the best performance.

moss, and abandoned tanks becoming "whale vertebrae." This transforms the theme of desolation into a cinematic, sensory experience. In contrast, the baseline relies on generic clichés such as "withered branches" and "a single blade of grass," failing to evoke unique emotional resonance.

Thematic Progression: Our model builds a subtle, organic progression from absolute barrenness to defiant rebirth. It trusts metaphor and surprise (e.g., "time's downy fur is growing") to convey the theme. Conversely, the baseline model explicitly spells out the moral in a didactic manner, ending with a slogan-like refrain ("emblem of humanity / society / future") that diminishes poetic ambiguity.

From a user experience perspective, the RL-generated poem successfully guides the reader's imagination and invites multiple readings, whereas the baseline output functions more as a lecture that evaporates upon first contact. This confirms that our RL training effectively incentivizes the model to move beyond safe, common patterns toward higher-quality, creative expression.

F The Use of Large Language Models (LLMs)

In this work, we used LLMs solely as a grammar and style assistant at the word and sentence level to polish writing. Specifically, we employed an LLM to double-check grammar and improve sentence-level readability, while ensuring that the core content in the paper, like ideation and experiments, was entirely developed by the authors.

User Instruction

I hope you can help me create a modern poem with the theme of 'Desolation and Hope' ... Use a stanza-based structure, no more than 8 lines per stanza; The language should be concise and powerful, rich in lyricism and vivid imagery ... emphasized contrast: symbolic scenes of desolation → elements of hope → enlightening philosophical sublimation.

Writing-RL (Ours): ***Spring in the Cracks***

I
Grains of sand count tree-rings in the wind
The earth splits open a thousand wounds
Migratory birds' flight paths erased by sandstorms
The ancient pendulum froze in 1972
Rusted iron bends into a crescent moon
Every scratch an unhealed scar
Only salt crystals, stubborn in the sunset
crystallize into vast white fields of salt

II
From a fissure a tender shoot forces its way
like a forgotten letter suddenly sprouting
...
Through gaps in corrugated roofs
moss weaves green longitude and latitude
with tiny root-hairs
—time's downy fur is growing

III
Migratory wings carry clouds
...
Abandoned oil tanks become whale vertebrae
vines climb the pipelines
weaving silver waterfalls under moonlight
...

IV
...
When the first spring wind sweeps the barren plain
all the dormant seeds
begin writing their own rings of years
—this birth inside the fissures
is the earth's deepest breath

Baseline (SFT): ***Desolation and Hope***

I
Withered branches sway in the wind,
a boundless desolate land.
Dry riverbeds, cracked earth,
a deathly silence, no trace of life.

II
Yet right there in that barren waste,
a single blade of grass tenaciously grows.
It sways in the wind,
stands tall beneath the scorching sun.
In desolation it seeks hope,
in despair it searches for life.

III
That desolate land
was once a thriving city.
Skyscrapers rose, traffic roared,
now only ruins remain.
But amid those ruins,
people have begun to rebuild their homes.
...

IV
Land of desolation, land of hope—
emblem of life itself.
Land of desolation, land of hope—
emblem of humanity.
Land of desolation, land of hope—
emblem of society.
Land of desolation, land of hope—
emblem of the future yet to come.

Figure 6: Comparison of generated poems. While the baseline SFT model relies on repetitive phrasing and explicit moralizing, our Writing-RL model employs original imagery ("rusty iron bends into a crescent moon," "green longitude and latitude") and achieves a sophisticated philosophical sublimation without didacticism.