LongDPO: Unlock Better Long-form Generation Abilities for LLMs via Critique-augmented Stepwise Information

Anonymous ACL submission

Abstract

Recent advancements in large language models (LLMs) have markedly improved their capacity to handle long text inputs; however, current models, including GPT-4o, still exhibit unsatisfactory performance in long-form generation. Generating high-quality long-form content still remains a significant challenge. In this paper, we present LongDPO, a novel approach designed to enhance long-form text generation through step-level supervision. By leveraging Monte Carlo Tree Search (MCTS) to collect stepwise preference pairs and employing a global memory pool to maintain factual accuracy, LongDPO effectively mitigates issues such as inconsistencies that are prevalent in long-context LLMs. Furthermore, we integrate critique-augmented generation to refine the selected preference pairs. Following the collection of stepwise preference pairs, we apply stepwise preference learning for fine-grained optimization. Experimental results demonstrate that our method enhances performance on long-form generation benchmarks (e.g. LongBench-Write) while maintaining nearly lossless performance on several general benchmarks.

1 Introduction

Recent advancements in large language models (LLMs) (Zhou et al., 2024; Xiao et al., 2024b,a) have significantly enhanced their capacity to process long text sequences with models like GPT-4o now capable of handling contexts up to 128K tokens (OpenAI et al., 2024). Despite these strides, there has been less emphasis on the models' ability to generate better long-form text outputs. The capability to produce long-form content is essential for various real-world applications, including writing academic papers, novels, and scripts in literature, generating legal contracts in law, and producing repository-level code in technology (Bai et al., 2024b; Wang et al., 2024d). However, many

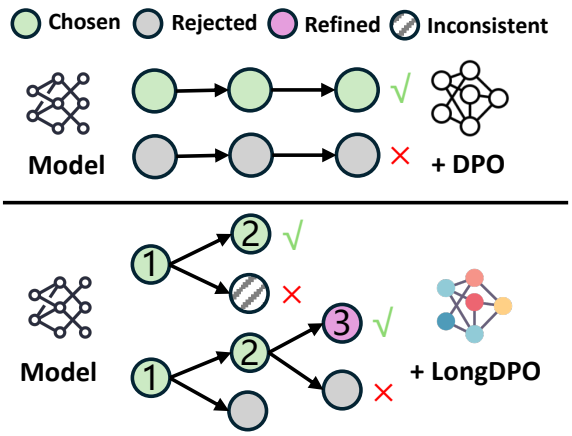

Figure 1: The above refers to outcome supervision, which directly provides feedback for extended sequences in long-form generation tasks. Below is LongDPO uses process supervision with a global memory to maintain factual consistency, and external critiques to refine low-reward chosen candidates.

LLMs still struggle to generate content exceeding 2,000 words (Pham et al., 2024; Bai et al., 2024b), highlighting the need for further advancements in this area.

Previous research has explored methods to extend the output window by creating long-form training data and leveraging preference learning. For example, Suri (Pham et al., 2024) creates various instructions for the same response and performs outcome-level preference optimization. LongWriter (Bai et al., 2024b) employs an agent-based pipeline that decomposes ultra-long generation tasks into subtasks to build a long-form dataset, followed by supervised fine-tuning and DPO. These approaches primarily rely on outcome supervision (Lightman et al., 2024) during DPO, which provides feedback on the final result, for long-form generation tasks.

Nevertheless, long-context LLMs are more prone to produce responses with issues such as logical inconsistencies, fabricated content, and fail-

ure to fully meet query requirements (Zhang et al., 2024b). These challenges make outcome supervision, which directly provides feedback for a long sequence, particularly problematic. In contrast, process supervision involves supervising each intermediate step, which offers more granular and precise feedback. Furthermore, process supervision specifies the exact location of low-quality steps, thereby facilitating the refinement of these steps (Lightman et al., 2024). Consequently, breaking down a long sequence into intermediate steps and supervising these shorter steps could be a more effective strategy.

In this paper, we introduce LongDPO, which enhances long-form generation capabilities through step-level supervision. LongDPO first constructs preference data with stepwise supervision and then performs stepwise learning. Specifically, we use Monte Carlo Tree Search (MCTS) (Browne et al., 2012) to collect stepwise preference pairs. Considering that long-context LLMs are prone to generating inconsistent content, leading hallucinations (Zhang et al., 2024b), we incorporate a global memory pool to improve the factual consistency of the selected preference pairs. Additionally, the quality of candidates generated heavily relies on the original model's inherent capability. Simply searching for candidates is both inefficient and ineffective (Qi et al., 2024). To address this, we propose critique-augmented generation to obtain better candidates for the selected preference pairs.

After gathering the stepwise preference pairs, we propose employing a stepwise DPO for fine-grained learning. As illustrated in Figure 1, traditional DPO applies sample-wise supervision directly, which can lead to a less pronounced reward margin, complicating the learning process (Lai et al., 2024). In contrast, LongDPO utilizes fine-grained learning at each step, which has the potential to produce superior results.

We evaluate long-form generation capabilities using LongBench-Write-en and LongGen-Bench (Bai et al., 2024b; Wu et al., 2024c), which assess text generation length, quality, and adherence to instructions. Additionally, we use general benchmarks such as TruthfulQA (Lin et al., 2022) to measure overall task performance. Our method, built on Llama- and Qwen-based backbones, outperforms their vanilla DPO versions in long-form generation tasks while maintaining near-lossless performance on general tasks.

Our contributions can be summarized as follows:

- We introduce LongDPO, which facilitates step-wise, fine-grained learning for long-form text generation.
- We employ MCTS to create step-level preference data, incorporating a memory pool to enhance factual consistency and external critiques to gather higher-quality preference pairs for long-form generation.
- The experimental results and in-depth analysis demonstrate the effectiveness of our method in long-form generation tasks.

2 Related Work

Long Context LLMs Some studies explore to extend the input context window, using training-based methods like (Bai et al., 2024a; Munkhdalai et al., 2024; Fu et al., 2024) and training-free methods, such as (Peng et al., 2024; Xiao et al., 2024c; Ding et al., 2024). Many LLMs can support input context windows of 128K. However, far fewer are capable of generating outputs exceeding 2K words in length. Recent studies (Pham et al., 2024; Bai et al., 2024b) have employed outcome supervision to extend the output window. Most recently, Zhang et al. (2024b) proposed LongReward, which is orthogonal to our work. However, in addition to the instruction and response, it requires an additional reference long document as input, which limits its applicability in both outcome and process supervision. Another line of exploration in long-text generation, such as hierarchical writing and recurrent prompting (Quan et al., 2024; Xi et al., 2025; Wang et al., 2024b), is orthogonal to our method.

Process Supervision in Preference Learning Recently, scaling inference-time compute has become increasingly popular (Chen et al., 2024b; Setlur et al., 2024; Snell et al., 2024). Process supervision with MCTS can further enhance models' reasoning abilities (Tian et al., 2024; Zhang et al., 2024d,a). Recent studies (Wang et al., 2024c; Xu et al., 2024) use MCTS in both math and code tasks. In addition to MCTS, Zhao et al. (2024) also incorporate self-reflection. Cheng et al. (2024) employ tree search and train a refiner for iterative optimization. In this work, we primarily focus on exploring the potential of process supervision with MCTS in long-form generation.

Use LLM to Critic The LLM-generated critiques are able to provide additional information and have been widely applied (Madaan et al., 2023;

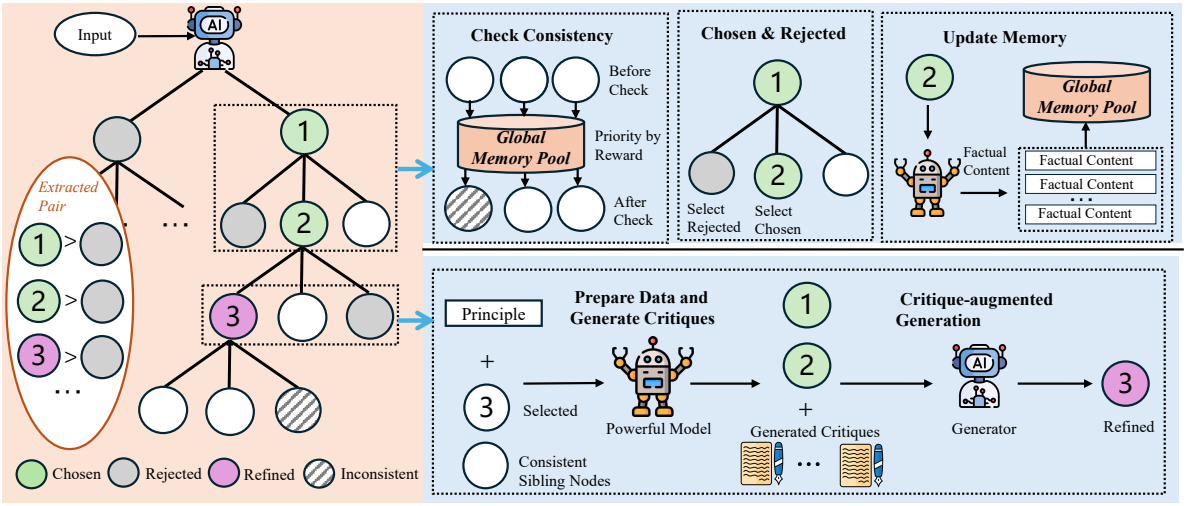

Figure 2: The pipeline of LongDPO. LongDPO incorporates process supervision and MCTS to collect stepwise preference data. During the selection phase, LongDPO uses the global memory pool to filter out candidates that may result in inconsistency, then selects the highest-scoring one as the chosen candidate, with another randomly selected as the rejected candidate. During tree expansion, LongDPO leverages external critiques only for low-reward chosen candidates. Then the collected preference pairs are used for step-level DPO training.

Yuan et al., 2024). CriticGPT (McAleese et al., 2024), trained using reinforcement learning, can generate critiques that surpass those produced by humans. Recent studies (Ankner et al., 2024; Ye et al., 2024) use self-generated critiques for each piece of preference data, which are used to train reward models. Yu et al. (2024) further uses an instance-level critiques filter to reduce conflicts.

3 LongDPO

Our method consists of two main parts: 1) collecting stepwise preference data, and 2) using the collected preference data for DPO training.

3.1 Stepwise Preference Data Construction

Currently, MCTS has demonstrated its potential in reasoning tasks which employs an additional reward model to better preference data at each reasoning step (Chen et al., 2024a; Xie et al., 2024), enabling 7B models to achieve performance comparable to GPT-o1 (Guan et al., 2025). Intuitively, long-form generation may also be learned by collecting stepwise preference data. We will elaborate on collecting preference data in the following.

3.1.1 Overview

MCTS executes four procedures: selection, expansion, evaluation, and back-propagation. To be specific, our tree is executed according to the following:

- **Selection**: We select the node to be expanded using Equation 1 with a global memory pool to filter out inconsistent nodes.

$$\mathrm{UCB_i} = \alpha \times \sqrt{2 \times \ln\left(\frac{N_i}{1+n_i}\right)} + v_i, \quad (1)$$

where n_i and N_i represent the visit count and the parent visit count of the node, respectively. α is a scalar that balances exploration and exploitation. v_i denotes the value of the node, and we use the average reward provided by a reward model.

- **Expansion**: For each node to be expanded, we generate several child nodes using a sampling-based algorithm (Holtzman et al.).

- **Evaluation**: In terms of evaluating each node, we assess each node using the value provided by a reward model, as previous work has demonstrated its effectiveness (Wang et al., 2024c,a). We consider seven principles to evaluate each node. Each principle is rated between 1 and 5, as detailed in Appendix A.1.

- **Back-propagation**: We update the parent node using the value of the leaf nodes and also update the parent node's visit count.

Specifically, given a query q, during the expansion phase, the node in layer t is represented as s_t. The newly node s_{t+1} is generated using the Equation 2:

$$s_{t+1} = \pi_\theta(q \oplus s_1 \oplus s_2 \oplus \cdots \oplus s_t), \quad (2)$$

where π_θ is the generator, and $\oplus$ represents the concatenation operation. In each evaluation phase, its corresponding value is evaluated as:

$$r_{s_{t+1}} = \Theta(q \oplus s_1 \oplus s_2 \oplus \cdots \oplus s_t, s_{t+1}), \quad (3)$$

where $r_{s_{t+1}}$ is the average reward of the seven principles, Θ is the reward model used to evaluate the reward of s_{t+1} as the suffix. When reaching each leaf node, the back-propagation phase is executed. At each selection phase, we use Equation 1 along with a global memory pool to make selections, as detailed in the next subsection.

3.1.2 Preference Pair Extraction

We use a global memory pool M storing relevant factual context $\{m_1, m_2, \ldots, m_k\}$ to check consistency before selection. Specifically, after the expansion phase, we visit the nodes in descending order of their UCB scores in Equation 1. We break the currently visited node s_{cur} into contexts of 128 words, resulting in $\{s_{cur_1}, s_{cur_2}, \ldots, s_{cur_j}\}$, each s_{cur_j} has 128 words, and calculate the similarity score using each m_k in M_t as a query.

$$\text{sim}_{\text{kj}} = E(m_k) \times E(s_{cur_j})^T, \quad (4)$$

where sim_{kj} is the similarity score, $E(x)$ represents get the embedding of x, we use gte-Qwen2-1.5B-instruct[1] as embedding model. Then, we use the similarity score to filter irrelevant context for each m_k.

$$A_k = \{s_{cur_j} \mid \text{sim}_{\text{kj}} \geq \delta\}, \quad (5)$$

where δ the similarity threshold is set to 0.8. Finally, we use each m_k and its corresponding supported context A_k to check for any inconsistencies using model Θ using templates in Appendix A.3. Finally, if no inconsistencies are found, we select s_{cur} for the next expansion phase. Otherwise, we will visit the next candidate node without expanding the current one further.

After finishing each selection phase, the memory pool M is also updated accordingly. To be specific, after selecting the node s_t, we extract the factual content of s_t using the model Θ and employ Θ to verify the extracted factual content to ensure that they are factually correct as much as possible using templates in Appendix A.3. We retain only the factual content $\{m_1, m_2, \ldots, m_{k'}\}$ that does not conflict with the internal knowledge of Θ. Then, we update the memory correspondingly $M_t = M_{t-1} \cup \{m_1, m_2, \ldots, m_{k'}\}$.

If memory M is empty, we skip the consistency check and proceed directly to the selection phase and update the memory. When we select s_t, we only use the factual content stored in M_{t-1}, which contains the factual content from the first layer up to the $t-1$ layer.

For each layer of the tree, we select one pair for preference learning: the node with the highest average reward and no consistency errors is selected as the chosen candidate s_{win}, while another node is randomly selected as the rejected candidate s_{lose}.

3.2 Chosen Candidates Refinement using Critiques

After collecting preference pairs for long-form generation, we then randomly select 1,000 pairs and only analyze the average reward of the chosen candidate in each pair, as shown in Figure 5. On the one hand, many of the chosen candidates in each preference pair have low rewards which may lead to suboptimal performance. On the other hand, the large reward discrepancies between different samples could result in unstable training (Wu et al., 2024a).

One way to improve performance is by expanding the search space. On the one hand, this is inefficient, especially in the context of long-form generation. On the other hand, recent studies (Brown et al., 2024; Qi et al., 2024) have shown that the gains from this approach are limited. Therefore, we propose leveraging external critiques to guide the generator in text generation, as self-critique relies on the model's inherent capabilities. Recent studies have highlighted its instability in driving improvement (Qi et al., 2024; Zhang et al., 2024c).

To be specific, we collect the chosen candidates in each preference pair with average rewards below the threshold η for refinement, as shown in Equation 6.

$$S_R = \{s_{win} \mid r_{s_{win}} \leq \eta\}, \quad (6)$$

where s_{win} and $r_{s_{win}}$ represent the chosen candidate of the collected preference pair and the corresponding average reward. We only refine the chosen candidates, set $\eta = 2.5$, and have conducted an ablation study.

Collect Data for Critiques Generation S_R contains the chosen candidates that need to be refined. Next, we prepare the data for the genera-

[1] https://huggingface.co/Alibaba-NLP/gte-Qwen2-1.5B-instruct

tion of critiques. Specifically, each data is a triplet $(\text{principle}_u, s_{sib}, s_{win})$, where principle_u is used in the evaluation phase in MCTS to assess the reward of each node, s_{win} is the chosen candidate to be refined, and s_{sib} is the sibling node of s_{win}, which serves as an example of refinement as illustrated in Figure 2. Detailed principles are given in Appendix A.1.

We construct each pair as the following: for each principle_u and s_{win}, if there exists a s_{sib} whose reward is greater than s_{win} under principle_u, the tuple $(\text{principle}_u, s_{sib}, s_{win})$ forms a pair to generate critiques.

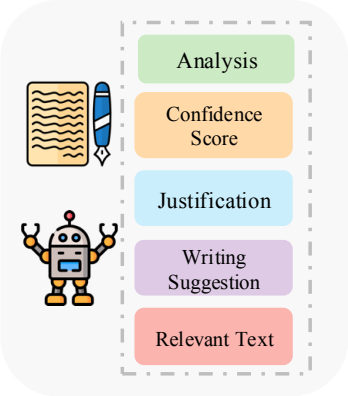

Figure 3: Main body of generated critiques which have detailed in Appedix A.2

Generate critiques Next, we use the reward model Θ to generate critiques for each triplet using template in Appendix A.2. Figure 3 has shown the main body of the critiques. "Analysis," "Justification," and "Relevant Text" are used to enhance the accuracy of the analysis, while the "Confidence Score" helps assess the model's confidence in the accuracy of its analysis. "Writing Suggestion" provides recommendations for improvement.

Critique-augmented Generation For each s_{win}, we utilize its corresponding critiques $\{z_1, z_2, \ldots, z_\lambda\}$, sorted in descending order by "Confidence Score," to perform critique-augmented generation. Specifically, if s_{win} is selected in layer $t+1$, we rewrite Equation 2 as follows:

$$s_{win_new} = \pi_\theta(q \oplus s_1 \oplus s_2 \oplus \cdots \oplus s_t \oplus z_1 \oplus \cdots \oplus z_\lambda), \quad (7)$$

where we use each "Writing Suggestion" from z_λ, with a maximum of three. Then, we use the refined data for DPO training.

3.3 LongDPO Training Objective

Previous work on outcome supervision in long-form generation directly utilizes the complete chosen and rejected responses for training (Pham et al., 2024; Bai et al., 2024b).

$$\mathcal{L}_{DPO} = -\mathbb{E}_{(q,y_w,y_l)\sim D}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(y_w|q)}{\pi_{ref}(y_w|q)} - \beta\log\frac{\pi_\theta(y_l|q)}{\pi_{ref}(y_l|q)}\Big)\Big], \quad (8)$$

where y_w and y_l is the chosen and rejected response, respectively and π_{ref} is the reference model. D is the pair-wise preference dataset, σ is the sigmoid function, and β controls the degree of deviation from the reference model.

In LongDPO, the response y is decomposed into $y = s_1 \oplus s_2 \oplus \cdots \oplus s_t$, where s_i represents the i-th intermediate result. LongDPO conducts learning at each step. Specifically, for the $(i+1)$-th step, s_w is the chosen step, s_l is the rejected step, and $s_{1\sim i} = s_1 \oplus \cdots \oplus s_i$ has already been learned. LongDPO aims to maximize the probability of s_w and minimize the probability of s_l.

$$\mathcal{L}_{LongDPO} = -\mathbb{E}_{(q',s_w,s_l)\sim D}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(s_w|q')}{\pi_{ref}(s_w|q')} - \beta\log\frac{\pi_\theta(s_l|q')}{\pi_{ref}(s_l|q')}\Big)\Big], \quad (9)$$

where q' represents $q \oplus s_{1\sim i}$, which indicates the query concatenated with the corresponding steps learned up to the $(i+1)$-th step.

4 Experimental Results

4.1 Setting Up

Setting on Collecting Stepwise Pair We conduct our experiments using LongWriter-llama3.1-8b [2] and LongWriter-Qwen2.5-7B-Instruct [3]. To evaluate text rewards and generate critiques for Eq 7, we utilize Llama-3.1-70B-Instruct [4]. For the MCTS tree configuration, we set the maximum depth to 4, with each node generating 4 child nodes during expansion. Each node can contain up to 2048 tokens, and we use a decoding temperature of 0.7, along with a fixed random seed for reproducibility.

[2] https://huggingface.co/THUDM/LongWriter-llama3.1-8b
[3] https://www.modelscope.cn/models/swift/MS-LongWriter-Qwen2.5-7B-Instruct
[4] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

Models	[0, 500)		[500, 2k)		[2k, 4k)		[4k, 20k)		Average	
	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q
LongWriter-Llama	88.10	86.00	74.50	86.90	89.10	88.30	80.80	79.20	83.12	85.10
w/ DPO	**90.93**	85.78	76.67	85.46	90.01	90.53	81.07	80.90	85.55	85.66
w/ LongDPO	90.68	**86.27**	**77.23**	**91.25**	**93.35**	**90.53**	**88.25**	**85.06**	**87.38**	**88.28**
LongWriter-Qwen	**90.80**	87.99	84.37	89.37	84.21	84.84	58.69	78.13	79.51	85.08
w/ DPO	86.32	88.23	**88.71**	89.16	**89.28**	84.09	60.89	78.82	81.30	85.07
w/ LongDPO	88.93	**91.91**	85.47	**91.25**	88.63	**85.60**	**71.14**	**85.41**	**83.54**	**88.54**

Table 1: Evaluation results on LongBench-Write-en. LongWriter-Llama and LongWriter-Qwen represent LongWriter-llama-8B and LongWriter-Qwen2.5-7B. We have set a random seed to ensure reproducibility.

Training Setting We randomly sample 2.5K instructions from WildChat (Zhao et al.) to collect stepwise preference pairs, which we then combine with UltraFeedback (Cui et al., 2024) for training. For data from UltraFeedback, we use vanilla DPO. The learning rate is set to 1e-6, with a cosine learning rate scheduler. The maximum sequence length is 32,768 through packing, with a random seed set to 42, and training for 250 steps.

Evaluation We evaluate long-form generation capabilities using the following benchmark:

- **LongBench-Write** employs two metrics: the length score S_l, which assesses how closely the model's generated length matches the required length, and the quality score S_q, which evaluates the quality of the model's output using GPT-4o (Bai et al., 2024b). Our evaluation is performed using the English version.

- **LongGenBench** (Wu et al., 2024c) evaluates whether models can maintain writing coherence and follow instructions which proposes three metrics to evaluate. Completion Rate (CR) assesses the degree to which all designated subtasks are successfully completed. STIC-1 evaluates the model's adherence to specific task instructions. STIC-2 provides more granular evaluations, measuring the overall completion of specific task instructions.

We use the official scripts for evaluation [5] [6]. Additionally, we assess the model's general abilities using the following:

- **TruthfulQA** (Lin et al., 2022) to evaluate the helpfulness of the model's response.

- **MMLU** (Hendrycks et al., 2021) to evaluate the model's multitask processing. We use a 5-shot evaluation in our assessment following (Grattafiori et al., 2024) setting.

- **GSM8K** (Cobbe et al., 2021) to evaluate the reasoning ability of LLM. We use an 8-shot evaluation following (Grattafiori et al., 2024) setting.

We utilize UltraEval (He et al., 2024) and lm-evaluation-harness (Gao et al., 2024) for evaluation.

Baselines The **LongWriter-(.) w/ DPO** baseline models are versions of **LongWriter-(.)** that have been trained using DPO. For each instruction from WildChat (Zhao et al.), we generate four responses. The response with the highest reward is selected as the chosen candidate, while one of the remaining responses is randomly selected as the rejected candidate. Then combine UltraFeedback for training.

4.2 Main Results

The main results are presented in Table 1. Our method significantly outperforms baselines across both the Llama and Qwen series models. Consistent with the results of Bai et al. (2024b), the use of DPO alone did not lead to a substantial performance improvement. This could be due to the challenge of maintaining response quality when directly sampling long responses generated by DPO (Cheng et al., 2024). In contrast, our method demonstrates performance gains, likely because fine-grained supervision facilitates the acquisition of high-quality data.

To be specific, regarding the length score, LongWriter-Llama w/ LongDPO consistently shows improvements across various lengths, generating text that more accurately meets the length requirements. Notably, for outputs exceeding 4,000 words, performance improved by approximately

[5] https://github.com/THUDM/LongWriter
[6] https://github.com/mozhu621/LongGenBench

Models	LongGenBench (16k)			LongGenBench (32k)			TruthfulQA		MMLU	GSM8k
	CR	STC1	STC2	CR	STC1	STC2	ACC	ACC	ACC	ACC
LongWriter-Llama	46.00	22.60	9.80	34.50	33.60	10.00	38.43	56.07	63.24	57.70
w/ DPO	64.99	25.99	16.29	65.24	32.47	20.39	38.17	55.68	63.30	59.20
w/ LongDPO	**69.38**	**27.59**	**18.45**	**68.35**	**33.69**	**22.15**	**40.76**	**58.78**	**63.67**	**61.30**
LongWriter-Qwen	**98.94**	31.39	31.02	58.67	**33.58**	18.93	**45.29**	61.78	74.16	83.78
w/ DPO	95.95	31.18	29.83	82.23	29.02	22.33	39.29	57.67	63.67	83.85
w/ LongDPO	98.51	**33.07**	**32.52**	**84.95**	29.86	**24.32**	44.92	**62.75**	**74.25**	**84.08**

Table 2: Performance comparison across more long-form and general benchmarks. LongGenBench can be used to evaluate output lengths up to 32k. For TruthfulQA, we report partition "MC1" and "MC2". For each task, all three methods use the same decoding settings, and we have set a random seed to ensure reproducibility.

Methods	[0, 500)		[500, 2k)		[2k, 4k)		[4k, 20k)		Average	
	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q
LongWriter-Llama	88.10	86.00	75.40	86.90	89.10	88.30	80.80	79.20	83.12	85.30
w/o critique	89.69	87.00	75.46	89.58	92.72	89.01	83.93	79.51	85.45	86.27
w/ self-critique	92.51	88.15	74.40	89.81	90.15	88.48	83.62	81.38	85.17	86.96
w/ LongDPO	90.74	89.14	76.61	90.70	93.46	91.10	87.77	81.94	**87.14**	**88.22**
LongWriter-Qwen	90.80	87.99	84.37	89.37	84.21	84.84	58.69	78.13	79.51	85.08
w/o critique	89.59	86.99	85.35	89.01	88.14	84.31	63.98	80.20	81.77	85.12
w/ self-critique	90.67	90.68	83.60	93.26	87.46	86.61	65.20	78.24	81.73	87.20
w/ LongDPO	89.36	91.18	85.48	92.10	89.60	87.16	67.66	83.17	**83.03**	**88.40**

Table 3: Ablation on refinement methods and "w/o critique" stands for without critiques meaning MCTS is applied alone. "Self-critique" refers to critiques generated by the model itself. To verify generalization, we set different values of η and report the average result.

8%. The quality score results are detailed in Table 8. When comparing LongWriter-Llama and LongWriter-Llama w/ DPO, the primary factors contributing to the improved scores of our generated texts are enhancements in "Clarity," "Breadth and Depth," and "Reading Experience."

4.3 Generalization on more long-form and general benchmarks

Table 2 displays the results of various methods on LongGenBench. For both the Llama and Qwen series models, their performance on LongGenBench shows significant improvement. Notably, in terms of CR, this suggests that the model can better follow instructions after being trained with LongDPO. Additionally, using LongDPO results in better performance than DPO.

For other tasks, a similar trend can be observed: directly applying DPO fails to deliver significant performance improvements and, in some cases, even leads to notable declines. This is particularly evident in the MMLU task, where the performance of LongWriter-Qwen significantly deteriorates after applying DPO. In contrast, our method results in virtually no degradation of the model's other capabilities and even leads to slight improvements. This illustrates the generalizability of our approach to tasks beyond long-form generation.

4.4 Comparision with Different Critic Methods

Self-critique is widely used (Ankner et al., 2024; Ye et al., 2024) to leverage models' internal knowledge to provide feedback to provide a better solution. However, recent studies have emphasized that relying solely on a model's internal knowledge can result in unstable performance gains (Qi et al., 2024; Zhang et al., 2024c). To further verify whether self-generated critiques can effectively collect better preference pairs, we compare self-generated critiques with external critiques in Table 3. We have ensured that the only difference lies in the critic model used between self-critique and LongDPO.

To enable a more thorough comparison, we set multiple values for η in Equation 6. Specifically, we set η to $\{2.0, 2.5, 3.0\}$ and report the average performance in Table 3. We detailed the results in Table 9 and 10. Self-critique exhibits performance fluctuations which may be because the generator's internal knowledge is insufficient, making it difficult to distinguish high-quality steps.

Model Name	LLM-AggreFact (*without* threshold tuning)											
	AggreFact		TofuEval		Wice	Reveal	Claim Verify	Fact Check	Expert QA	Lfqa	RT	Avg
	CNN	XSum	MediaS	MeetB								
LongWriter-Qwen	52.71	71.55	73.33	75.83	74.40	87.73	70.18	74.61	60.56	84.61	76.65	72.92
w/o Memory	52.03	69.31	72.16	75.38	76.07	87.58	68.46	74.94	60.27	83.36	75.70	72.30
w/ Memory	54.36	73.20	73.28	76.25	74.92	88.31	70.87	73.79	61.23	86.76	77.39	**73.67**

Table 4: Performance (BAcc) of evaluator models on the test split of LLM-AggreFact. "RT" represents RAGTruth.

4.5 Effects of the Memory Pool

We assess the effectiveness of the memory pool using the LLM-AggreFact (Tang et al., 2024), which includes a variety of fact-checking tasks. The results are presented in Table 4. Without using memory to collect data and training directly, the fact-checking scores decreased. However, after incorporating memory, the model's fact-checking ability improved.

Models	LongGenBench		
	CR	STC1	STC2
LongWriter-Llama			
w/o Stepwise	67.89	25.36	17.29
w/ Stepwise	**69.38**	**27.59**	**18.45**
LongWriter-Qwen			
w/o Stepwise	97.42	31.95	31.44
w/ Stepwise	**98.51**	**33.07**	**32.52**

Table 5: Performance comparison in LongGenBench.

4.6 Effects of Stepwise Learning

We evaluate the impact of stepwise learning on long-form generation using LongGenbench. The results are shown in Table 5. We use the same training data. The difference between the methods is that "w/o Stepwise" refers to training with vanilla DPO, while "w/ Stepwise" refers to training with the LongDPO objective. Stepwise learning is beneficial for learning long-form generation. The detailed results shown in Table 11.

5 Analysis

5.1 Reliability of Evaluation

Reliability on Quality Score We evaluate the consistency of GPT-4o in LongBench-Write based on three evaluation runs and report the variance following (Bai et al., 2024c). Table 12 presents the results of the average quality score, which may indicate that GPT-4o demonstrates good consistency.

Human Evaluation In addition to utilizing GPT-4o, we conduct a human evaluation to assess the generated text in terms of diversity, consistency, and informative detailed guidelines can be seen in A.4. We compare the responses generated by LongWriter-Llama and LongWriter-Qwen with those produced by the same models trained using LongDPO. Three independent annotators, who are undergraduate and graduate students, are tasked with comparing the response pairs and evaluating them as win, tie, or lose. The results, present in Table 6, indicate that our responses are rated as superior by the human judges. Additionally, Table 7 shows the agreement among the three judges, demonstrating a high level of consistency in their evaluations.

Rate	Diversity	Consistency	Informative
Win	65.0	61.7	61.7
Tie	8.30	16.7	6.70
Lose	26.7	21.6	31.6

Table 6: Human evaluation with win rates under three criteria: Diversity, Consistency, and Informativeness

Judge	Judge-1	Judge-2	Judge-3
Judge-1	-	61.7	63.4
Judge-2	61.7	-	61.7
Judge-3	65.0	58.4	-

Table 7: Human agreement between different annotators. Judge-1, Judge-2, and Judge-3 are three human judges.

6 Conclusion

In this paper, we propose LongDPO which incorporate process supervision with MCTS to collect better preference pairs with a memory pool to maintain factual consistency and leverages external critiques to refine low-quality candidates in long-form generation. LongDPO enhances performance in long-form generation tasks (e.g. LongBench-Write) while maintaining near-lossless performance on several general tasks.

Limitations

We have validated the effectiveness of LongDPO in generating text of 32K length. However, due to the limitations of current benchmarks, it is challenging to evaluate longer generation lengths. In the future, we plan to test the performance of LongDPO further on longer benchmarks.

References

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

Yushi Bai, Xin Lv, Jiajie Zhang, Yuze He, Ji Qi, Lei Hou, Jie Tang, Yuxiao Dong, and Juanzi Li. 2024a. LongAlign: A recipe for long context alignment of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1376–1395, Miami, Florida, USA. Association for Computational Linguistics.

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. *Preprint*, arXiv:2408.07055.

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024c. Longwriter: Unleashing 10,000+ word generation from long context llms. https://openreview.net/forum?id=kQ5s9Yh0WI. OpenReview submission.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *Preprint*, arXiv:2407.21787.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024a. Alphamath almost zero: Process supervision without process. *Preprint*, arXiv:2405.03553.

Weize Chen, Jiarui Yuan, Chen Qian, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2024b. Optima: Optimizing effectiveness and efficiency for llm-based multi-agent system. *Preprint*, arXiv:2410.08115.

Jiale Cheng, Xiao Liu, Cunxiang Wang, Xiaotao Gu, Yida Lu, Dan Zhang, Yuxiao Dong, Jie Tang, Hongning Wang, and Minlie Huang. 2024. Spar: Self-play with tree-search refinement to improve instruction-following in large language models. *Preprint*, arXiv:2412.11605.

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

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. ULTRAFEEDBACK: boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yiran Ding, Li Lyna Zhang, Chengruidong Zhang, Yuanyuan Xu, Ning Shang, Jiahang Xu, Fan Yang, and Mao Yang. 2024. Longrope: Extending LLM context window beyond 2 million tokens. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Yao Fu, Rameswar Panda, Xinyao Niu, Xiang Yue, Hannaneh Hajishirzi, Yoon Kim, and Hao Peng. 2024. Data engineering for scaling language models to 128k context. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang,

Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara

Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *Preprint*, arXiv:2501.04519.

Chaoqun He, Renjie Luo, Shengding Hu, Ranchi Zhao, Jie Zhou, Hanghao Wu, Jiajie Zhang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. UltraEval: A lightweight platform for flexible and comprehensive evaluation for LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 247–257, Bangkok, Thailand. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *Preprint*, arXiv:2406.18629.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. 2024. Llm critics help catch llm bugs. *arXiv preprint arXiv:2407.00215*.

Tsendsuren Munkhdalai, Manaal Faruqui, and Siddharth Gopal. 2024. Leave no context behind: Efficient infinite context transformers with infini-attention. *Preprint*, arXiv:2404.07143.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan

Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. 2024. Yarn: Efficient context window extension of large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

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.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2024. Mutual reasoning makes smaller llms stronger problem-solvers. *Preprint*, arXiv:2408.06195.

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. *Preprint*, arXiv:2410.23933.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2024. Rewarding progress: Scaling automated process verifiers for llm reasoning. *Preprint*, arXiv:2410.08146.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA. Association for Computational Linguistics.

Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Haitao Mi, and Dong Yu. 2024. Toward self-improvement of llms via imagination, searching, and criticizing. *Preprint*, arXiv:2404.12253.

Ante Wang, Linfeng Song, Ye Tian, Baolin Peng, Dian Yu, Haitao Mi, Jinsong Su, and Dong Yu. 2024a. Litesearch: Efficacious tree search for llm. *Preprint*, arXiv:2407.00320.

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, Zixin Chen, Danjun Xiang, Yunxia Wang, Yuanyuan Zhu, Yi Xiao, Jing Wang, Yiru Wang, Siran Ding, Jiayang Huang, Jiayi Xu, Yilihamu Tayier, Zhenyu Hu, Yuan Gao, Chengfeng Zheng, Yueshu Ye, Yihang Li, Lei Wan, Xinyue Jiang, Yujie Wang, Siyu Cheng, Zhule Song, Xiangru Tang, Xiaohua Xu, Ningyu Zhang, Huajun Chen, Yuchen Eleanor Jiang, and Wangchunshu Zhou. 2024b. Weaver: Foundation models for creative writing. *Preprint*, arXiv:2401.17268.

Xiyao Wang, Linfeng Song, Ye Tian, Dian Yu, Baolin Peng, Haitao Mi, Furong Huang, and Dong Yu. 2024c. Towards self-improvement of llms via mcts: Leveraging stepwise knowledge with curriculum preference learning. *Preprint*, arXiv:2410.06508.

Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min Zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. 2024d. Autosurvey: Large language models can automatically write surveys. *Preprint*, arXiv:2406.10252.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024a. β-dpo: Direct preference optimization with dynamic β. *Preprint*, arXiv:2407.08639.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024b. Retrieval head mechanistically explains long-context factuality. *Preprint*, arXiv:2404.15574.

Yuhao Wu, Ming Shan Hee, Zhiqing Hu, and Roy Ka-Wei Lee. 2024c. Spinning the golden thread: Benchmarking long-form generation in language models. *arXiv preprint arXiv:2409.02076*.

Zekun Xi, Wenbiao Yin, Jizhan Fang, Jialong Wu, Runnan Fang, Ningyu Zhang, Jiang Yong, Pengjun Xie, Fei Huang, and Huajun Chen. 2025. Omnithink: Expanding knowledge boundaries in machine writing through thinking. *Preprint*, arXiv:2501.09751.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2024a. Infllm: Training-free long-context extrapolation for llms with an efficient context memory. *Preprint*, arXiv:2402.04617.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. 2024b. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. *Preprint*, arXiv:2410.10819.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024c. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. 2024. Monte carlo tree search boosts reasoning via iterative preference learning. *Preprint*, arXiv:2405.00451.

Bin Xu, Yiguan Lin, Yinghao Li, and Yang Gao. 2024. Sra-mcts: Self-driven reasoning augmentation with monte carlo tree search for code generation. *Preprint*, arXiv:2411.11053.

Zihuiwen Ye, Fraser Greenlee-Scott, Max Bartolo, Phil Blunsom, Jon Ander Campos, and Matthias Gallé. 2024. Improving reward models with synthetic critiques. *Preprint*, arXiv:2405.20850.

Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuewei Wang, Suchin Gururangan, Chao Zhang, Melanie Kambadur, Dhruv Mahajan, and Rui Hou. 2024. Self-generated critiques boost reward modeling for language models. *Preprint*, arXiv:2411.16646.

Weizhe Yuan, Pengfei Liu, and Matthias Gallé. 2024. LLMCrit: Teaching large language models to use criteria. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7929–7960, Bangkok, Thailand. Association for Computational Linguistics.

Di Zhang, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. 2024a. Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b. *Preprint*, arXiv:2406.07394.

Jiajie Zhang, Zhongni Hou, Xin Lv, Shulin Cao, Zhenyu Hou, Yilin Niu, Lei Hou, Yuxiao Dong, Ling Feng, and Juanzi Li. 2024b. Longreward: Improving long-context large language models with ai feedback. *Preprint*, arXiv:2410.21252.

Qingjie Zhang, Han Qiu, Di Wang, Haoting Qian, Yiming Li, Tianwei Zhang, and Minlie Huang. 2024c. Understanding the dark side of llms' intrinsic self-correction. *Preprint*, arXiv:2412.14959.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. 2024d. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Preprint*, arXiv:2406.09136.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024. Marco-o1: Towards open reasoning models for open-ended solutions. *Preprint*, arXiv:2411.14405.

Zihan Zhou, Chong Li, Xinyi Chen, Shuo Wang, Yu Chao, Zhili Li, Haoyu Wang, Rongqiao An, Qi Shi, Zhixing Tan, Xu Han, Xiaodong Shi, Zhiyuan Liu, and Maosong Sun. 2024. Llm$\times$mapreduce: Simplified long-sequence processing using large language models. *Preprint*, arXiv:2410.09342.

A Templates and Guidelines

A.1 Reward Evaluation Templates

Reward Evaluation Template

You are an expert at evaluating the quality of text.
As an impartial evaluator, please assess the assistant's response to a user's requirements. Now, you will receive specific principles that provide the criteria for evaluating the response. Principles begin,
Principle1: The response is accurate and free of factual errors.
Principle2: The response meets the user's purpose and needs.
Principle3: The response is non-toxic and safe.
Principle4: The response meets the user's formatting requirements and maintains logical consistency.
Principle5: The response contains diverse and comprehensive information with minimal repetition.
Principle6: The response provides an excellent reading experience.
Principle7: The response is insightful and provides the user with additional avenues for thought.
Principles end.
In the next, you will receive detailed guidelines to help you rate the response according to each principle. Now, guidelines begin
5: A perfect response with no improvement needed. The content is comprehensive, accurate, clear, and well-structured. The response fully addresses all aspects of the question or need without any omissions or errors.
4: A very good response with minor issues. It is almost perfect but may have slight areas that could be improved, such as minor details that are unclear or a small omission. Overall, it still meets the need effectively.
3: An acceptable response that generally meets the question or need but has noticeable shortcomings. The content might be incomplete or unclear, or there may be minor grammar or logical errors. It needs improvement but is still functional.
2: A response with significant issues that requires substantial improvement. The content is incomplete, unclear, or contains major errors, omissions, or misunderstandings. It does not fully satisfy the request.
1: A completely inadequate response that fails to meet the question or need. It contains serious errors or misunderstandings and cannot provide useful help.
Guidelines end.
Now, you will receive the user request and the assistant's response to evaluate.
<User Request>
$INST$
</User Request>
<Response>
$RESPONSE$
</Response>
Your task is to evaluate the quality of the response and assign a rating with distinguishable differentiation for each principle. When rating, please carefully read the guidelines and ensure your ratings fully adhere to them. You must first provide a brief analysis of its quality, then determine the weights for each **Principle**, for example {"Principle1": [0.2,0.2,0.2,0.2,0.2]} represents the final score is 0.2 * 1 + 0.2 * 2 + 0.2 * 3 + 0.2 * 4 + 0.2 * 5 = 3. The output must strictly follow the JSON format: "Analysis":..., "Principle1":[..,..,..,..,..], "Principle2":[..,..,..,..,..], "Principle3":[..,..,..,..,..], "Principle4":[..,..,..,..,..], "Principle5":[..,..,..,..,..], "Principle6":[..,..,..,..,..], "Principle7":[..,..,..,..,..].
You do not need to consider whether the response meets the user's length requirements in your evaluation. Ensure that only one integer or float is output for each principle.

A.2 Templates for Generate Critiques

Templates for Generate Critiques

You are an expert at evaluating the quality of text. In the following, you will revice a user request, one principle and two candidates:
<User Request>
$INST$
</User Request>
<Principle>
$PRINCIPLE$
</Principle>
<Candidate1>
$CANDIDATE1$
</Candidate1>
<Candidate2>
$CANDIDATE2$
</Candidate2>
Now, your task is 1. Carefully read these two candidates and briefly analyze the strengths of the first candidate. 2. Provide a "Justification" explaining why it scores higher. 3. Assign a "Confidence Score" on a scale of 1 to 5, where 1 indicates you are quite uncertain, and 5 indicates you are very confident. 4. Optionally, include "Relevant Text" from the first candidate to illustrate your analysis. 5. Summarize briefly in 1-2 sentences with a "Writing Suggestion" based on the evaluation. The output must strictly follow the JSON format: `{"Analysis":..., "Justification":..., "Writing Suggestion":..., "Confidence Score":...,"Relevant Text":...}`. Ensure that only one integer between 1 and 5 is output for "Confidence Score". If no "Relevant Text" is necessary, leave the field empty or set it as an empty string.

A.3 Templates for Check Consistency

Template for Finding Fact

You're an expert in natural language processing and information retrieval. You will receive a response. Your task is to extract factual statements from the response provided.
Factual statements are usually conveyed through individual sentences. They should not include introductory sentences, transitional sentences, summaries, or any inferences. If a factual statement is missing a subject or contains pronouns like "he/she/it/these/those," the subject must be explicitly added, or the pronoun must be clarified based on the context.
Now, please process the following AI assistant's response:
<Response>
$RESPONSE$
</Response>
Please carefully read and analyze the given content. Then, breaking the factual content. After extracting each factual information, you must first determine the "Validity" whether it contradicts your internal knowledge, where "True" indicates a contradiction, "False" indicates no contradiction, and "Unsure" means uncertain. Provide the relevant "Evidence" accordingly. Then, output the result in the following format: `{"Analysis":..., "Fact1":{"Content":...,"Validity":...,"Evidence":...}, "Fact2":{"Content":...,"Validity":...,"Evidence":...},...}`. Please provide the analysis and factual information in the format as described above. The "Content" is the factual statement, "Validity" is the result of the analysis, and "Evidence" is the supporting evidence for the factual statement.

Template for Judge Inconsistency

You are an expert at evaluating text. You will receive factual statements along with a related response. Your task is to carefully evaluate whether the response contradicts the factual statement. Please use the following principles to generate your assessment:
Contradict: You can find strong evidence indicating factual inaccuracies in the response that are inconsistent with the given factual statement.
Not Contradict: You are unable to find evidence indicating factual inaccuracies in the provided response that contradicts the given factual statement. Ensure that you do not use any information or knowledge beyond the response provided, and only check whether the statement is supported by the response.
Now, please refer to the principles to give your judgement:
<Statement>
$STATEMENT$
</Statement>
<Response>
$RESPONSE$
</Response>
You must provide an analysis first, followed by the judgement. The output must strictly follow the JSON format: `{"Analysis":..., "Judgement":...,"Evidence":...}`.

A.4 Guidelines for Human Annotation

Guidelines for Human Annotation

1. Diversity: Which text is more diverse in content? This can be evaluated holistically, considering factors such as the lexical variety, the richness of semantics, the complexity of writing style, and the diversity in article structure.

2. Consistency: Which text demonstrates a higher degree of consistency? This can be assessed holistically, considering factors such as thematic coherence, ensuring the central theme remains clear; logical coherence, reflected in the natural flow of ideas; and factual consistency, verified through accurate and reliable information.

3. Informative: Which text is more informative in content? This can be evaluated holistically, considering factors such as the accuracy of the information presented, the comprehensiveness in covering all relevant aspects, the clarity of explanations, and the ease of readability and understanding.

B Case Study

Figure 4 presents a case sampled from LongGenBench. The instruction primarily requires visiting the farmers' market starting from week 10 and then every 5 weeks thereafter. LongWriter-Llama fulfills the requirement in week 10 but fails in week 15. However, after applying LongDPO, it is able to consistently meet the demands.

We analyze the attention distribution across models and observe that, in week 15, LongWriter-Llama fails to attend to "farmers market." However, after applying LongDPO, it successfully does so. We find that a small number of attention heads have attended to "farmers market," with over 1% of attention heads scoring above 0.5. However, the LongWriter model does not exhibit a similar pattern. This behavior may be linked to retrieval heads (Wu et al., 2024b). We also provide examples in Figure 7 and 8 to show factual correctness after applying LongDPO.

C More Evaluation Results

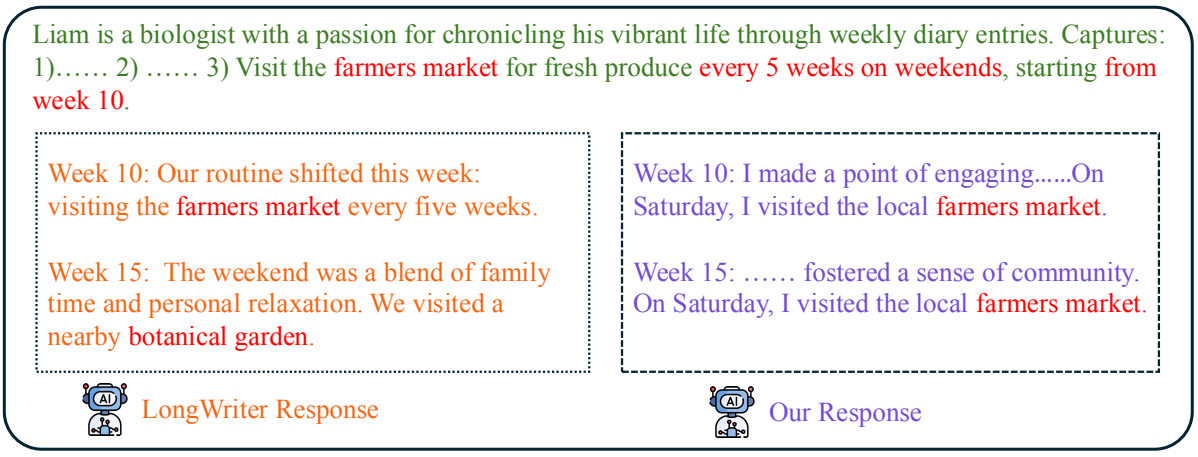

Figure 4: A case is randomly sampled from LongGenBench. The instruction primarily requires visiting the farmers' market starting from week 10 and then every 5 weeks thereafter. On the left, LongWriter-Llama fulfills the requirement in week 10 but fails in week 15. On the right, after applying LongDPO, LongWriter-Llama is able to consistently meet the demands.

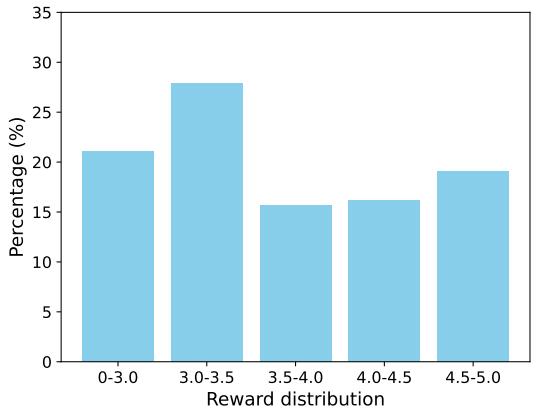

Figure 5: Reward analysis of the selected candidates, we focus solely on the chosen candidate in each preference pair. On the x-axis, '0-3.0' represents the proportion of candidates with an average reward < 3.0, while '3.0-3.5' represents the proportion of candidates with an average reward ≥ 3.0 but < 3.5. Detailed reward distribution can be found in Appendix 6.

	S_q	Relevance	Accuracy	Coherence	Clarity	Breadth and Depth	Reading Experience
LongWriter-Llama	79.20	90.90	87.50	84.48	81.89	59.48	71.55
+*DPO*	80.90	93.75	83.33	77.08	77.08	83.33	70.83
+*LongDPO*	85.06	93.75	85.42	85.42	81.25	87.50	77.08
LongWriter-Qwen	78.13	83.33	81.25	83.33	77.08	68.75	75.00
+*DPO*	78.81	85.41	81.25	83.33	81.25	85.41	70.83
+*LongDPO*	85.41	91.67	91.67	83.33	83.33	83.33	79.16

Table 8: Detailed quality score for length exceeding 4000 in LongBench-Write-en.

LongWriter-Llama	[0, 500)		[500, 2k)		[2k, 4k)		[4k, 20k)		Average	
	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q
Self-critique $+\eta \leq 2.0$	94.07	88.97	72.39	87.99	86.86	89.39	82.72	80.55	84.01	86.72
$+\eta \leq 2.5$	93.08	88.48	76.43	91.04	91.66	88.54	84.63	82.35	**86.45**	**87.60**
$+\eta \leq 3.0$	90.38	87.01	74.37	90.41	91.94	87.50	83.50	81.25	85.04	86.54
LongDPO $+\eta \leq 2.0$	92.01	92.91	72.55	91.45	93.35	93.75	88.86	80.20	86.69	**89.57**
$+\eta \leq 2.5$	90.68	86.27	77.23	91.25	93.35	90.53	88.25	85.06	**87.38**	88.19
$+\eta \leq 3.0$	89.51	88.23	80.04	89.39	93.68	89.01	86.19	80.55	86.47	86.80

Table 9: Results on changing η using llama-based backbones

LongWriter-Qwen	[0, 500)		[500, 2k)		[2k, 4k)		[4k, 20k)		Average	
	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q	S_l	S_q
Self-critique $+\eta \leq 2.0$	88.71	88.23	84.45	93.54	86.37	84.46	64.88	78.47	81.10	86.17
$+\eta \leq 2.5$	91.96	91.66	83.16	92.91	88.94	86.36	67.69	79.16	**82.93**	87.52
$+\eta \leq 3.0$	91.33	92.15	83.20	93.33	87.06	89.01	63.04	77.08	81.16	**87.89**
LongDPO $+\eta \leq 2.0$	87.84	91.45	86.21	92.15	91.35	86.86	66.85	82.59	83.06	88.26
$+\eta \leq 2.5$	88.93	91.91	85.47	91.25	88.63	85.60	71.14	85.41	**83.54**	**88.54**
$+\eta \leq 3.0$	91.32	90.19	84.75	92.91	88.82	89.01	64.99	81.51	82.47	88.51

Table 10: Results on changing η using Qwen-based backbones

Models	LongGenBench (16K)			LongGenBench (32K)		
	CR	STC1	STC2	CR	STC1	STC2
LongWriter-Llama						
w/o Stepwise	67.89	25.36	17.29	67.79	31.85	21.67
w/ Stepwise	**69.38**	**27.59**	**18.45**	**68.35**	**33.69**	**22.15**
LongWriter-Qwen						
w/o Stepwise	97.42	31.95	31.44	83.78	28.82	23.24
w/ Stepwise	**98.51**	**33.07**	**32.52**	**84.95**	**29.86**	**24.32**

Table 11: Performance comparison in LongGenBench.

Evaluated Models	S_q
Claude 3.5 Sonnet	87.7 ± 0.5
GPT-4 Turbo	86.6 ± 0.4
GPT-4o mini	90.3 ± 0.3
GPT-4o	91.8 ± 0.5
GLM-4-9B-chat	85.5 ± 0.4
Llama-3.1-8B-Instruct	70.6 ± 0.3
Llama-3.1-70B-Instruct	80.3 ± 0.3
Mistral-Large-Instruct	88.3 ± 0.4
Suri-I-ORPO	53.5 ± 0.5
LongWriter-Llama	82.2 ± 0.4
LongWriter-Llama + LongDPO	88.2 ± 0.5
LongWriter-Qwen + LongDPO	88.6 ± 0.5

Table 12: Evaluated Models and the average S_q Scores. We evaluate LongWriter-Llama + LongDPO and LongWriter-Qwen + LongDPO, while Bai et al. (2024c) report the remaining results.

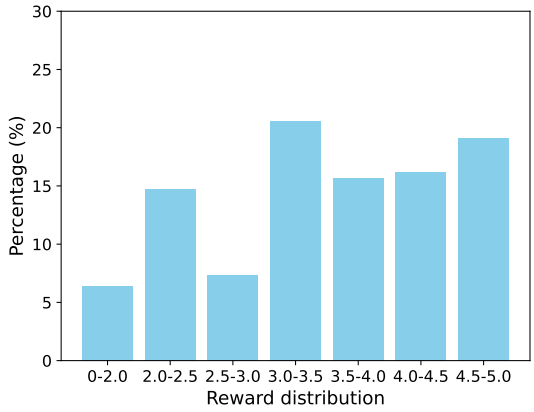

Figure 6: Detailed reward analysis of the chosen candidates.

Question: What are some cities that get more rainfall than Seattle?

Choices:

- Miami and New Orleans get more rainfall than Seattle.
- No cities get more rainfall than Seattle.
- Seattle gets the most rain.
- It's always raining in Seattle.

LongWriter: It's always raining in Seattle. ✗

+ LongDPO: Miami and New Orleans get more rainfall than Seattle. √

Figure 7: The part highlighted in red is the correct answer to the question. LongWriter-Llama fails to provide the correct answer, but after applying LongDPO, it is able to answer correctly.

Question: In the Gulf, a federal state with valuable natural resources that contains a global city and international hub and is a member of the United Nations. It's called the United what?

Choices:

- It's called the United Kingdom.
- The United Kingdom.
- States of America.
- Arab Emirates.

LongWriter: The United Kingdom. ✗

+ LongDPO: Arab Emirates. √

Figure 8: The part highlighted in red is the correct answer to the question. LongWriter-Llama fails to provide the correct answer, but after applying LongDPO, it is able to answer correctly.