Textual Aesthetics in Large Language Models

Lingjie Jiang[1,2*], **Shaohan Huang**[1,✉], **Xun Wu**[1], **Furu Wei**[1]
[1]Microsoft Research [2]Peking University
lingjiejiang@stu.pku.edu.cn; {shaohanh, xunwu, fuwei}@microsoft.com

Abstract

Image aesthetics is a crucial metric in the field of image generation. However, textual aesthetics has not been sufficiently explored. With the widespread application of large language models (LLMs), previous work has primarily focused on the correctness of content and the helpfulness of responses. Nonetheless, providing responses with textual aesthetics is also an important factor for LLMs, which can offer a cleaner layout and ensure greater consistency and coherence in content. In this work, we introduce a pipeline for aesthetics polishing and help construct a textual aesthetics dataset named TEXAES. We propose a textual aesthetics-powered fine-tuning method based on direct preference optimization, termed TAPO, which leverages textual aesthetics without compromising content correctness. Additionally, we develop two evaluation methods for textual aesthetics based on text and image analysis, respectively. Our experiments demonstrate that using textual aesthetics data and employing the TAPO fine-tuning method not only improves aesthetic scores but also enhances performance on general evaluation datasets such as AlpacalEval and Arena-hard. Our code and data are available at https://github.com/JackLingjie/Textual-Aesthetics.

1 Introduction

Image aesthetics (Huang et al., 2024a; Murray et al., 2012; Kong et al., 2016; Ke et al., 2021; Bosse et al., 2017) has emerged as a prominent research area within computer vision, focusing on assessing and improving the visual appeal of images. Aesthetics has recently been integrated into state-of-the-art image generation models, such as diffusion models (Rombach et al., 2022), significantly enhancing the visual quality of generated images (Wu et al., 2024a, 2023) and aligning them more closely with human preferences (Huang et al., 2024a; Wu et al., 2024b, 2023).

Meanwhile, advancements in large language models (LLMs) like ChatGPT (OpenAI, 2023) and LLaMA (Touvron et al., 2023a; Dubey et al., 2024) have demonstrated impressive generative capabilities across various domains, including code, articles, and web content. Although LLMs have made significant progress in generating textual content, enhancing the aesthetic quality of their output remains a critical challenge. A more aesthetically appealing and organized output not only improves user engagement by making the content more intuitive and comfortable to read but also enhances consistency and coherence. Consequently, exploring the textual aesthetics of LLMs is a highly desirable area of research.

In this work, we present the first investigation into improving the aesthetic quality of text generated by LLMs. Unlike image aesthetics benefiting from numerous large-scale aesthetic datasets (e.g., AVA (Murray et al., 2012) and AesBench (Huang et al., 2024b)), advanced aesthetic learning technology (Huang et al., 2024a; Zhang and Liu, 2023; Yang et al., 2022; Su et al., 2020) and reliable aesthetic evaluation methods (Deng et al., 2017; Su et al., 2011), textual aesthetics in LLMs lacks similar resources and established models.

To address this challenge, we first designed an aesthetic data generation pipeline leveraging GPT-4o for aesthetic polishing. This scalable pipeline can generate large volumes of high-quality aesthetic preference data. Based on this framework, we constructed the first aesthetic dataset in the LLM domain, TEXAES, which contains a total of 50,390 prompts data.

Based on TEXAES, existing post-training techniques such as DPO (Rafailov et al., 2024b) can be used to fine-tune current LLMs at the aesthetic level. However, we found that directly applying these techniques not only failed to align effectively

[*] Contribution during internship at Microsoft. [✉] Corresponding Author.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 13790–13818
November 4-9, 2025 ©2025 Association for Computational Linguistics

with the characteristics of our TexAes, limiting its impact on aesthetic fine-tuning, but also negatively impacted the overall performance of these LLMs. To address this issue, we propose **T**extual **A**esthetics **P**reference **O**ptimization (TAPO) which employs the Plackett-Luce (Luce, 1959; Plackett, 1975) model with adjustable optimization weights to better leverage our dataset and enhance aesthetic fine-tuning performance. Furthermore, to better assess the aesthetic quality of LLM outputs, we have developed two evaluation pipelines: one based on text and the other based on images, respectively.

To validate the effectiveness of our TexAes and TAPO, we performed aesthetic fine-tuning on the open-source LLaMA series models (Dubey et al., 2024) and compared the aesthetic scores of the fine-tuned LLMs with state-of-the-art LLMs at different scales (from 7B to 72B). Additionally, to ensure objective and reliable results, we employed human experts for professional evaluation. Extensive experimental results ultimately demonstrated the effectiveness of our TexAes and TAPO.

Our main contributions are listed as follow:

- To the best of our knowledge, we for the first time indicate the crucial issue of exploring and improving the textual aesthetics in LLMs.
- We systematically identify the lack of related textual aesthetics datasets, and introduce a novel pipeline for aesthetic text polishing and contribute to the construction of a textual aesthetics dataset, named TexAes.
- Based on TexAes, we propose a DPO-based aesthetic fine-tuning algorithm, named TAPO, to effectively enhance the LLMs' aesthetic quality while preserving its general performance.
- Both qualitative and quantitative extensive experiments demonstrate that utilizing TexAes and TAPO not only improves aesthetic scores but also enhances the general capabilities of LLMs.

2 Related Works

2.1 Image Aesthetics

Image aesthetics (Huang et al., 2024a; Murray et al., 2012; Kong et al., 2016) is a subfield of computer vision that focuses on assessing (Deng et al., 2017; Su et al., 2011) and improving the aesthetic quality of images (Bhattacharya et al., 2010; Deng et al., 2018). Early work in the field of image aesthetics focused on using handcrafted metrics to assess aesthetic scores (Nack et al., 2001; Neumann et al., 2005). However, with the development of deep learning, there has been significant interest in applying CNN (Bosse et al., 2017; Li et al., 2018; Su et al., 2020) or Transformer (Ke et al., 2021; Zhang and Liu, 2023; Yang et al., 2022; Qin et al., 2023) based methods to solve image aesthetics problems, which have demonstrated promising results. Recently, multi-modal large language models (MLLMs) have shown superior aesthetic perception and robustness in the fields of image aesthetics, greatly surpassing lightweight models due to their vast knowledge base and strong reasoning and memory capabilities (Huang et al., 2024a,b; Wu et al., 2024b).

2.2 LLM Preferences Data

Preference learning is an optimization method for LLMs designed to enhance their ability to generate outputs that better align with human preferences (Fürnkranz and Hüllermeier, 2010; Schulman et al., 2017; Rafailov et al., 2024b; Ouyang et al., 2022). Increasing attention has also been drawn to the importance of data used during the preference learning phase. Some studies focus on constructing domain-specific datasets for preference learning, e.g., summarization (Stiennon et al., 2020; Wu et al., 2021) and question answering (Nakano et al., 2021). Cui et al. (2024) highlight the scarcity of large-scale, general-purpose preference datasets and propose UltraFeedback to address this gap by collecting over 1 million preference feedback samples using GPT-4 (OpenAI, 2023). Lee et al. (2023) also point out that utilizing AI-generated preference feedback is an effective and cost-efficient method for expanding preference datasets. While the aforementioned work provides preference datasets for specific domains as well as general-purpose tasks, none of them have addressed the critical area of text aesthetics in LLMs, which motivated us to design corresponding data construction pipeline and related dataset like TexAes to support future research in text aesthetics.

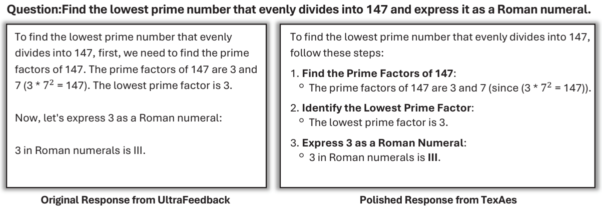
3 Textual Aesthetics

3.1 Overview

Textual aesthetics, which encompass the aesthetic attributes of a text at both the content and visual levels, can be dissected into four fundamental aspects. **Clarity** (readability) pertains to the ease with which a text can be read and comprehended, necessitating optimal sentence length and grammatical complexity (DuBay, 2004). **Layout** (visual organization) involves the systematic arrangement of text elements, such as headings and subheadings, to guide the reader effectively. **Uniformity** (consistency) demands a consistent style and formatting throughout the text to enhance readability and facilitate a smoother reading experience. **Coherence** (overall structure) ensures that paragraphs are well-organized and logically connected, facilitating easier comprehension of the content (Van Silfhout et al., 2014).

3.2 Aesthetics Polishing

Human preference data is critical for aligning large language models and improving their performance across various dimensions, such as helpfulness (Askell et al., 2021; Kreutzer et al., 2018; Stiennon et al., 2020), harmlessness (Bai et al., 2022; Glaese et al., 2022), and honesty (Ouyang et al., 2022). Consequently, we believe that a textual aesthetic preference dataset will also be beneficial for research on the alignment of LLMs. However, current literature reveals a conspicuous absence of research specifically addressing the textual aesthetics of LLMs, as well as a lack of corresponding textual aesthetic preference data. To address this gap, we have developed a method for textual aesthetic polishing to construct a dataset that optimizes the aesthetic preferences of LLMs.



Question:Find the lowest prime number that evenly divides into 147 and express it as a Roman numeral.

To find the lowest prime number that evenly divides into 147, first, we need to find the prime factors of 147. The prime factors of 147 are 3 and 7 (3 * 7^2 = 147). The lowest prime factor is 3.

Now, let's express 3 as a Roman numeral:

3 in Roman numerals is III.

Original Response from UltraFeedback

To find the lowest prime number that evenly divides into 147, follow these steps:

1. **Find the Prime Factors of 147**:
   - The prime factors of 147 are 3 and 7 (since (3 * 7^2 = 147)).
2. **Identify the Lowest Prime Factor**:
   - The lowest prime factor is 3.
3. **Express 3 as a Roman Numeral**:
   - 3 in Roman numerals is **III**.

Polished Response from TexAes

Figure 1: Comparison of responses between the UltraFeedback and TEXAES datasets.

Given that the goal of polishing is to enhance textual aesthetics, we can build our textual aesthetic preference dataset based on an available preference dataset such as UltraFeedback (Cui et al., 2024). UltraFeedback is a comprehensive dataset with responses evaluated by GPT-4 based on criteria such as instruction-following, honesty, and helpfulness. Since the selected data exhibits higher scores in these areas, thereby aligning more closely with human preferences, we can utilize these chosen responses as our candidates to build our textual aesthetic preference dataset.

To effectively achieve our objectives, we designed a chain of thought (Wei et al., 2023) methodology by using GPT-4o to polish our original responses, the following steps were taken:

1. **Semantic Analysis**: GPT-4o initially analyzed the textual semantics of the provided instructions and selected responses.
2. **Aesthetic Evaluation**: Based on textual aesthetic factors such as paragraph structure, indentation, headings, and subheadings, GPT-4o conducted a detailed textual aesthetic analysis.
3. **Binary Classification**: GPT-4o then performed a binary classification to determine whether the response required modification to improve readability and comprehension.
4. **Revision Process**: For responses that required modification, GPT-4o generated a revised version that preserved the original semantics while enhancing readability and comprehensibility.

Figure 1 displays the text before and after modification, demonstrating the enhanced readability and comprehension of the revised text. More cases can be found in Appendix F.1, and the polishing prompts are provided in Appendix G.1. Finally, we compiled the textual aesthetic preference dataset $\mathcal{D} = \left\{ \left(x^{(i)}, y_t^{(i)}, y_w^{(i)}, y_l^{(i)} \right) \right\}_{i=1}^{N}$, where $y_t^{(i)}$ represents the revised textual aesthetic data, and $y_w^{(i)}$ and $y_l^{(i)}$ represent the originally chosen and rejected data in UltraFeedback, respectively.

We observed that some polished responses became overly verbose and less natural or human-like. We hypothesize that this is because the original responses in UltraFeedback are already of high quality, making the task of polishing more challenging than expected. To address this issue, we implemented a length constraint for the polishing process. Future work will focus on further improving the textual aesthetic polishing method.

3.3 Textual Aesthetics Scoring

To validate the aesthetic quality of texts generated by large language models and to assess the effectiveness of our aesthetic preference dataset, a robust method for evaluating text aesthetics is indispensable. Previous studies, such as AlpacaEval (Li et al., 2023; Dubois et al., 2024), MT-Bench (Zheng et al., 2023), and Arena-Hard (Li et al., 2024), suggest that using LLMs as evaluators can effectively approximate human preferences. Consequently, we employ the "LLM-as-a-judge" framework to approximate human preferences for text aesthetics. We evaluate the aesthetic quality of texts generated by LLMs using two methods: text-based and image-based text aesthetics scoring.

Text-Based Text Aesthetics Scoring. We randomly selected 500 prompts from Arena-Hard (Li et al., 2024) as our evaluation dataset. Following practices of Arena-Hard and MT-Bench (Zheng et al., 2023), we implemented a pairwise comparison method, comparing the performance of model π_i on prompt p with a robust baseline model π_{base} (GPT-4-0314) to derive aesthetic preference scores. Judges assessed aesthetic preferences on a Likert scale (Likert, 1932) (1 = prefers $\pi_i(p)$ much less than $\pi_{\text{base}}(p)$, 5 = prefers $\pi_i(p)$ much more than $\pi_{\text{base}}(p)$). This methodology ensures that models are penalized more heavily for substantial losses than for minor ones, effectively differentiating between models. Using the chain-of-thought approach, judges evaluated text aesthetics based on four dimensions: readability, visual organization, consistency, and overall structure. To mitigate position bias, we employed a two-game setup by swapping model positions for each query. Following the practices of Chatbot Arena, we adopted the Bradley-Terry (Bradley and Terry, 1952) model to generate final scores. We aggregated all pairwise comparisons with the baseline model to derive win rates and used bootstrapping to calculate a confidence interval, resulting in an ordered ranking of all models. The judge prompts are provided in Appendix G.2.

Image-Based Text Aesthetics Scoring. Our conceptualization of text aesthetics encompasses not only textual readability and comprehensibility but also visual appeal. Given GPT-4o's exceptional multimodal capabilities, we utilized GPT-4o to evaluate text aesthetics from a visual perspective as well. In our experiments, we rendered the LLM-generated texts as HTML with consistent CSS styles, converted them into images of identical size, and then had GPT-4o evaluate these images based on the same criteria used for textual evaluation. Specific prompts are provided in Appendix G.3.

4 Textual Aesthetics-Powered Training

4.1 Direct Preference Optimization Training

Reinforcement Learning with Human Feedback (RLHF) (Christiano et al., 2017) has emerged as a pivotal technique in aligning LLMs (Bai et al., 2022; Ouyang et al., 2022; Stiennon et al., 2020). Early implementations of RLHF primarily relied on reinforcement learning and alternative approaches (Snell et al., 2022; Touvron et al., 2023b; Gulcehre et al., 2023). Rafailov et al. (2024a) proposed a RL-free closedform counterpart known as Direct Preference Optimization (DPO) which has shown impressive performances (Ivison et al., 2023; Jiang et al., 2023; Tunstall et al., 2023).

The naive DPO uses a pair of preference data, which includes a chosen response and a rejected response for each prompt, based on the Bradley-Terry (Bradley and Terry, 1952) model for optimization. The loss function for DPO is defined as follows:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log\sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right), \tag{1}$$

where π_θ denotes the policy being optimized, π_{ref} represents the reference policy, x is the input prompt, y_w is the chosen (winning) response, y_l is the rejected (losing) response, $\mathcal{D}$ is the dataset of prompts and responses, σ is the sigmoid function, and β is a scaling parameter. By directly integrating preference data into the optimization process, DPO ensures that the generated text aligns closely with human judgments.

4.2 Textual Aesthetics Preference Optimization Training

For each prompt in our TexAes dataset, there are three responses: y_t, y_w, and y_l. The response y_t has the same semantic content as y_w but is superior in terms of textual aesthetics. The response y_w, in turn, is more aligned with human preferences for chatbots in terms of instruction-following, truthfulness, honesty, and helpfulness compared to y_l. The goal of our training is to learn a model that can generate responses that are both aesthetically pleasing and preferred by humans. To achieve this, we designed a textual aesthetics preference optimization (TAPO) approach that jointly optimizes for both textual aesthetics and human preferences.

To simultaneously utilize all three preference data types in the TexAes for optimization, we adopt the Plackett-Luce (Luce, 1959; Plackett, 1975) model as the underlying preference model. We denote each reward function $\beta \log \frac{\pi_\theta(y_k|x)}{\pi_{\text{ref}}(y_k|x)}$ as $r_\theta(x, y_k)$, which Rafailov et al. (2024a) showed can be treated as an "implicit reward" representing the preference for the model to generate y_k given the prompt x (where $k \in \{t, w, l\}$). Here, π_θ and π_{ref} are the policy model and reference model, respectively, and β is the parameter controlling deviation from the reference model. The training objective of TAPO is

$$\mathcal{L}_{\text{TAPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_t,y_w,y_l)\sim\mathcal{D}} \log\left(\frac{\exp(r_\theta(x,y_t))}{\sum_{i\in\{t,w,l\}}\exp(r_\theta(x,y_i))} \cdot \frac{\exp(r_\theta(x,y_w))}{\sum_{i\in\{w,l\}}\exp(r_\theta(x,y_i))}\right) \tag{2}$$

Using the properties of logarithmic functions, the loss function can be decomposed into two parts: $\mathcal{L}_{\text{TA}}$ and $\mathcal{L}_{\text{DPO}}$:

$$\begin{aligned} \mathcal{L}_{\text{TA}} &= -\log\left(\frac{\exp(r_\theta(x,y_t))}{\sum_{i\in\{t,w,l\}}\exp(r_\theta(x,y_i))}\right), \\ \mathcal{L}_{\text{DPO}} &= -\log\left(\frac{\exp(r_\theta(x,y_w))}{\sum_{i\in\{w,l\}}\exp(r_\theta(x,y_i))}\right). \end{aligned} \tag{3}$$

It can be observed that $\mathcal{L}_{\text{DPO}}$ is identical to the loss used in Bradley-Terry model-based preference optimization with y_w and y_l, as demonstrated in the proof provided in Appendix C. On the other hand, $\mathcal{L}_{\text{TA}}$ represents the log probability of $r_\theta(x, y_t)$ being ranked first among $r_\theta(x, y_t)$, $r_\theta(x, y_w)$, and $r_\theta(x, y_l)$. $\mathcal{L}_{\text{DPO}}$ primarily optimizes the model's preference for honest, helpful, and truthful data, whereas $\mathcal{L}_{\text{TA}}$ optimizes both the correctness of the answers and textual aesthetics. To ensure the generated answers are not only accurate but also aesthetically pleasing, we assign different weights to the losses to adjust the preference optimization direction. The modified loss function is as follows:

$$\begin{aligned} &\mathcal{L}_{\text{TAPO}}(\pi_\theta, \pi_{\text{ref}}) = \\ &- \mathbb{E}_{(x,y_t,y_w,y_l)\sim\mathcal{D}}\left[w_{\text{TA}} \cdot \mathcal{L}_{\text{TA}} + w_{\text{DPO}} \cdot \mathcal{L}_{\text{DPO}}\right]. \end{aligned} \tag{4}$$

5 Data and Experiment Settings

5.1 Textual Aesthetics Dataset

As introduced in Section 3.2, we constructed our textual aesthetic dataset based on a filtered version of UltraFeedback[1] (Cui et al., 2024; Ivison et al., 2023; Tunstall et al., 2023) dataset, which comprises 61,135 completions evaluated by GPT-4, including both accepted and rejected entries. In our experiment, we utilized GPT-4o to perform aesthetic polishing on the UltraFeedback dataset. After the aesthetic polishing process, we found that 5,858 entries were already aesthetically satisfactory and required no further modification. We then analyzed the length of the filtered texts and discovered that a minor subset exhibited excessive verbosity and lacked a natural, human-like quality. To address this, we excluded outliers in the distribution of length differences before and after aesthetic polishing, retaining only data within the 90% confidence interval. We present the statistics of TEXAES in Table 1. The length distribution and the length constraint filter experiment are detailed in Appendix A and Appendix D.1, respectively.

Dataset	#Prompts	Response Length
ULTRAFEEDBACK	61,135	297
TEXAES	50,390	293

Table 1: Statistics of TEXAES datasets.

5.2 Experiment Settings

In this study, we evaluate the performance of models from two perspectives: textual aesthetics and general response capabilities. For textual aesthetics, we compare the models using both text-based and image-based text aesthetic scoring methods, as described in Section 3.3. We report the win rate (WR) in text aesthetics at both the text and image levels relative to the baseline model (GPT-4-0314). In addition to automatic evaluation, we conduct a human evaluation to further validate the models' performance. We randomly sample fifty entries from the Anera-Hard (Li et al., 2024) dataset and ask human annotators to rate the aesthetics of these entries.

To evaluate the changes in the model's general capabilities following the alignment of textual aesthetics preferences, including its ability to follow instructions and respond to complex prompts across diverse domains, we utilize three well-established auto-evaluation instruction-following benchmarks based on GPT-4-as-a-Judge: AlpacaEval 2.0 (Dubois et al., 2024), Arena-Hard (Li et al., 2024) and MT-Bench (Zheng et al., 2023). For both the supervised fine-tuning and TAPO stages, we employ a low-rank adaptation (Hu et al., 2021) adapter instead of fine-tuning the entire model. Detailed training parameters are provided in the Appendix B.

6 Experiment Results

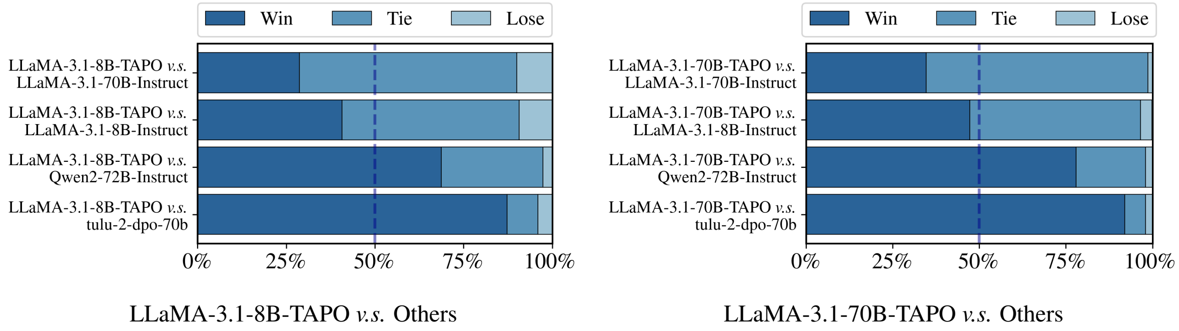
6.1 Main Results

The comparative analysis of our models trained with TAPO on TEXAES against open-source mod-

[1] https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized

Model	Size	TA Text WR(%)	TA Image WR(%)	AlpacaEval 2.0 LC WR(%)	Arena-Hard WR(%)	MT-Bench Avg. Score
Mistral-7B-Instruct-v0.3 (Jiang et al., 2023)	7B	8.26	28.90	29.87	17.13	6.59
Mistral-7B-Instruct-v0.3-TAPO	7B	28.55	57.84	38.53	23.10	6.80
Qwen2-7B-Instruct (Yang et al., 2024)	7B	24.63	39.40	33.43	27.69	7.48
Qwen2-7B-Instruct-TAPO	7B	37.99	64.28	40.27	32.40	7.48
Yi-1.5-9B-Chat (AI et al., 2024)	9B	35.52	55.03	34.74	**38.89**	7.38
LLaMA-3.1-8B-Instruct (Dubey et al., 2024)	8B	33.42	47.94	41.34	37.10	7.42
LLaMA-3.1-8B-TAPO	8B	**50.85**	**71.91**	**49.84**	33.89	**7.72**
Tulu-2-dpo-70B (Ivison et al., 2023)	70B	9.43	27.79	31.01	16.37	6.89
Qwen2-72B-Instruct (Yang et al., 2024)	72B	22.05	30.68	40.61	42.48	8.22
LLaMA-3.1-70B-Instruct (Dubey et al., 2024)	70B	53.18	57.34	45.03	**67.22**	8.16
LLaMA-3.1-70B-TAPO	70B	**63.22**	**73.31**	**51.26**	63.42	**8.30**

Table 2: Performance comparison between TAPO models and open-source models across various benchmarks."TA Text" and "TA Image" denote text-based and image-based textual aesthetic metrics, respectively. Metrics include: win rates (WR) against GPT-4-Turbo for TA Text and TA Image, WR against GPT-4-0314 for Arena-Hard, length-controlled win rate (LC WR) against GPT-4-Turbo in AlpacaEval 2.0, and average scores for MT-Bench. All evaluations are conducted using GPT-4o as the judge.



LLaMA-3.1-8B-TAPO *v.s.* Others

LLaMA-3.1-70B-TAPO *v.s.* Others

Figure 2: Win rates of models fine-tuned by TAPO compared to other SOTA open-source models by human judgements in textual aesthetics level. Human judgments are majority votes from three annotators.

els is shown in Table 2. Our LLaMA-3.1-70B-TAPO model surpasses all open-source counterparts in both text-based and image-based text aesthetic metrics, with an 18.88% improvement in text-based scores and a 27.85% enhancement in image-based scores over the best-performing LLaMA-3.1-70B-Instruct model.

For general response benchmarks, the LLaMA-3.1-8B-Instruct and LLaMA-3.1-70B-Instruct models, after TAPO training, show improvements on AlpacaEval 2.0 and MT-Bench, though with a slight decline on Arena-Hard. AlpacaEval 2.0 focuses on chat scenarios, MT-Bench on multi-turn conversations, and Arena-Hard on more complex queries. The gains in AlpacaEval 2.0 and MT-Bench suggest that enhanced text aesthetics contribute to better conversational abilities, aligning with our goal of improving answer clarity, layout, uniformity, and coherence. This underscores the quality of TexAes and the effectiveness of TAPO in boosting both text aesthetics and overall model performance. Furthermore, the results from experiments using TexAes and TAPO on Qwen2 (Yang et al., 2024) and Mistral (Jiang et al., 2023) demonstrate similar performance improvements, showcasing the generalizability of TAPO across diverse model architectures, as detailed in Appendix D.2.

The results of the human evaluation, shown in Figure 2, show that our LLaMA-3.1-70B-TAPO model is rated significantly higher in text aesthetics than the best-performing open-source model. These results confirm that the responses of our model are more visually appealing and coherent, consistent with our quantitative analysis, further validating the efficacy of TAPO in enhancing text aesthetics and overall performance.

6.2 Ablation Studies

To validate the efficacy of incorporating three types of preference data in TAPO, we conducted a two-stage DPO training ablation experiment. Initially, human preferences were aligned using the y_w and y_l data sets, denoted as DPO(y_w, y_l). Subsequently, text aesthetic preference alignment was conducted

Training Settings	TA Text WR(%)	TA Image WR(%)	AlpacaEval 2.0 LC WR(%)	Arena-Hard WR(%)	MT-Bench Avg. Score
LLaMA-3.1-8B-Base					
DPO(y_t, y_w)	15.72	51.70	15.78	4.10	5.19
DPO(y_t, y_l)	25.79	**60.64**	24.06	9.04	5.78
DPO(y_w, y_l) + DPO(y_t, y_w)	14.03	48.31	14.66	5.35	5.50
DPO(y_w, y_l) + DPO(y_t, y_l)	25.45	60.53	23.77	7.72	5.98
TAPO(y_t, y_w, y_l)	**25.61**	55.43	**26.05**	**9.16**	**6.05**
LLaMA-3.1-8B-Instruct					
DPO(y_t, y_w)	46.89	71.19	38.93	26.04	7.36
DPO(y_t, y_l)	49.07	68.63	45.82	29.87	7.55
DPO(y_w, y_l) + DPO(y_t, y_w)	50.76	**75.69**	44.91	29.41	7.39
DPO(y_w, y_l) + DPO(y_t, y_l)	50.26	71.33	46.47	31.08	**7.75**
TAPO(y_t, y_w, y_l)	**50.85**	71.91	**49.84**	**33.89**	7.72

Table 3: Comparison of two-stage DPO training and TAPO training. DPO(y_w, y_l) + DPO(y_t, y_l) denotes two-stage training where the first stage is DPO(y_w, y_l) and the second stage is DPO(y_t, y_l). Similarly, DPO(y_w, y_l) + DPO(y_t, y_w) follows this format.

using two methods: DPO(y_t, y_w) and DPO(y_t, y_l). These experiments were performed on the *LLaMA-3.1-Base* and *LLaMA-3.1-Instruct* models, where the *LLaMA-3.1-Base* model was first fine-tuned on the Tulu-v2 dataset (Ivison et al., 2023) in a supervised manner. The results are presented in Table 3.

When comparing the final models from the two-stage training with our model trained in TAPO method, we found that, except for the image-based text aesthetic metric, where our model was slightly inferior, it significantly outperformed the two-stage models on text-based aesthetic metrics, AlpacaEval 2.0, Arena-Hard, and MT-Bench. This suggests that TAPO, by leveraging three types of preference data, not only enhances text aesthetic scores but also improves general capabilities.

We further investigated the impact of the weight ratio between $\mathcal{L}_{\text{TA}}$ and $\mathcal{L}_{\text{DPO}}$ in TAPO on both the aesthetics and the overall performance of the generated texts. To this end, we conducted a series of systematic experiments. As detailed in Appendix D.3, the results further demonstrate the effectiveness of TAPO.

Finally, to validate the effectiveness of TEXAES, we conduct two ablation experiments: one comparing TEXAES and UltraFeedback with DPO training, as detailed in Appendix D.4, and another examining the criteria for selecting rejected samples and their impact on the model's performance, as discussed in Appendix D.5. The results from these experiments demonstrate the effectiveness of TEXAES in optimizing both text aesthetics and general performance.

6.3 Annotation Consistency

We generated responses for 50 questions sampled from Arena-Hard using six models: *LLaMA-3.1-8B-TAPO*, *LLaMA-3.1-70B-TAPO*, *LLaMA-3.1-8B-Instruct* (Dubey et al., 2024), *LLaMA-3.1-70B-Instruct* (Dubey et al., 2024), *Qwen2-72B-Instruct* (Yang et al., 2024), and *Tulu-2-dpo-70B* (Ivison et al., 2023). Subsequently, we employed three types of evaluators: text-based GPT-4o judge (TA Text), image-based GPT-4o judge (TA Image), and three human annotators (details can be found in Appendix E). Each evaluator was tasked with comparing *LLaMA-3.1-8B-TAPO* and *LLaMA-3.1-70B-TAPO* against other models in terms of the textual aesthetics of the generated answers (win/tie/lose), resulting in 400 annotated comparison pairs.

Judge	A-1	A-2	A-3	Average
A-1	-	78.25%	77.50%	77.88%
A-2	78.25%	-	80.75%	79.50%
A-3	77.50%	80.75%	-	79.13%
TA Image	60.75%	68.00%	65.75%	64.83%
TA Text	69.00%	70.00%	67.00%	68.67%

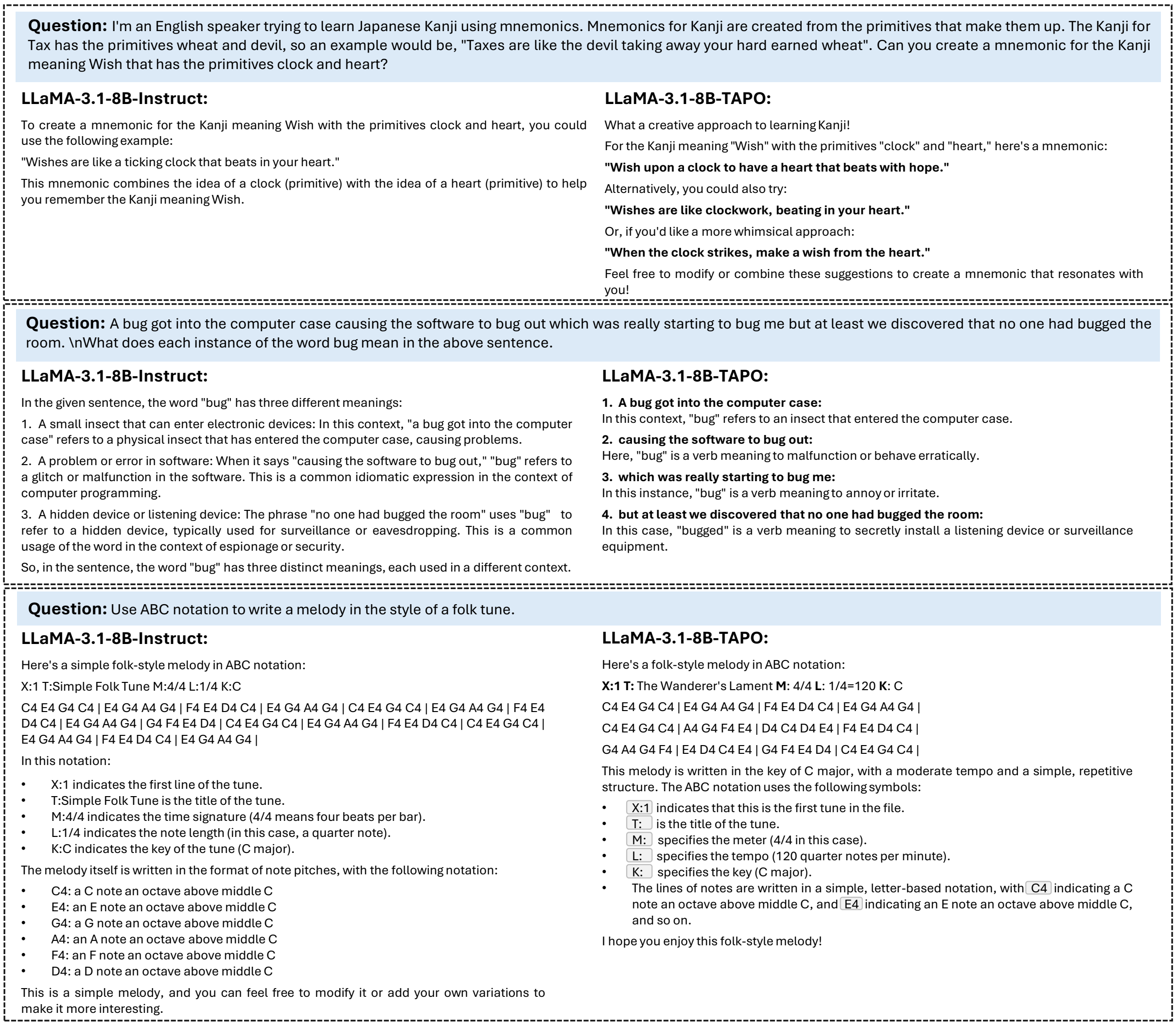
Table 4: Agreement between judges and human annotators on 400 samples from Arena-Hard. A-1, A-2, and A-3 are three human annotators. TA Text is the text-based GPT-4o judge, and TA Image is the image-based GPT-4o judge.

Table 4 presents the agreement ratios, as utilized in MT-Bench (Zheng et al., 2023), among the TA Text scores, TA Image scores, and annotators, as well as annotators themselves. On average, the TA Text scores demonstrated a 68.67% agreement rate with the human annotators, while the TA Image scores exhibited a 64.83% agreement rate, which is lower than that of the human annotators. Notably, the agreement rates of both our image-based and text-based GPT-4o judges are comparable to those observed in previous human evaluations, which reported an average of 66% agreement in MT-Bench (Zheng et al., 2023) and 59.7% in UltraFeedback (Cui et al., 2024). These results suggest that our GPT-4o judges can serve as effective proxies for human preferences in assessing text aesthetics.

7 Case Study

In this section, we compare *LLaMA-3.1-8B-Instruct* and *LLaMA-3.1-8B-TAPO* from three cases in Arena-Hard, as shown in Figure 3.

The first case (top of Figure 3) demonstrates how *LLaMA-3.1-8B-TAPO* improves the mnemonic for



Question: I'm an English speaker trying to learn Japanese Kanji using mnemonics. Mnemonics for Kanji are created from the primitives that make them up. The Kanji for Tax has the primitives wheat and devil, so an example would be, "Taxes are like the devil taking away your hard earned wheat". Can you create a mnemonic for the Kanji meaning Wish that has the primitives clock and heart?

LLaMA-3.1-8B-Instruct:

To create a mnemonic for the Kanji meaning Wish with the primitives clock and heart, you could use the following example:

"Wishes are like a ticking clock that beats in your heart."

This mnemonic combines the idea of a clock (primitive) with the idea of a heart (primitive) to help you remember the Kanji meaning Wish.

LLaMA-3.1-8B-TAPO:

What a creative approach to learning Kanji!

For the Kanji meaning "Wish" with the primitives "clock" and "heart," here's a mnemonic:

"Wish upon a clock to have a heart that beats with hope."

Alternatively, you could also try:

"Wishes are like clockwork, beating in your heart."

Or, if you'd like a more whimsical approach:

"When the clock strikes, make a wish from the heart."

Feel free to modify or combine these suggestions to create a mnemonic that resonates with you!

Question: A bug got into the computer case causing the software to bug out which was really starting to bug me but at least we discovered that no one had bugged the room. \nWhat does each instance of the word bug mean in the above sentence.

LLaMA-3.1-8B-Instruct:

In the given sentence, the word "bug" has three different meanings:

1. A small insect that can enter electronic devices: In this context, "a bug got into the computer case" refers to a physical insect that has entered the computer case, causing problems.
2. A problem or error in software: When it says "causing the software to bug out," "bug" refers to a glitch or malfunction in the software. This is a common idiomatic expression in the context of computer programming.
3. A hidden device or listening device: The phrase "no one had bugged the room" uses "bug" to refer to a hidden device, typically used for surveillance or eavesdropping. This is a common usage of the word in the context of espionage or security.

So, in the sentence, the word "bug" has three distinct meanings, each used in a different context.

LLaMA-3.1-8B-TAPO:

1. A bug got into the computer case:
In this context, "bug" refers to an insect that entered the computer case.

2. causing the software to bug out:
Here, "bug" is a verb meaning to malfunction or behave erratically.

3. which was really starting to bug me:
In this instance, "bug" is a verb meaning to annoy or irritate.

4. but at least we discovered that no one had bugged the room:
In this case, "bugged" is a verb meaning to secretly install a listening device or surveillance equipment.

Question: Use ABC notation to write a melody in the style of a folk tune.

LLaMA-3.1-8B-Instruct:

Here's a simple folk-style melody in ABC notation:

X:1 T:Simple Folk Tune M:4/4 L:1/4 K:C

C4 E4 G4 C4 | E4 G4 A4 G4 | F4 E4 D4 C4 | E4 G4 A4 G4 | C4 E4 G4 C4 | E4 G4 A4 G4 | F4 E4 D4 C4 | E4 G4 A4 G4 | G4 F4 E4 D4 | C4 E4 G4 C4 | E4 G4 A4 G4 | F4 E4 D4 C4 | C4 E4 G4 C4 | E4 G4 A4 G4 | F4 E4 D4 C4 | E4 G4 A4 G4 |

In this notation:

- X:1 indicates the first line of the tune.
- T:Simple Folk Tune is the title of the tune.
- M:4/4 indicates the time signature (4/4 means four beats per bar).
- L:1/4 indicates the note length (in this case, a quarter note).
- K:C indicates the key of the tune (C major).

The melody itself is written in the format of note pitches, with the following notation:

- C4: a C note an octave above middle C
- E4: an E note an octave above middle C
- G4: a G note an octave above middle C
- A4: an A note an octave above middle C
- F4: an F note an octave above middle C
- D4: a D note an octave above middle C

This is a simple melody, and you can feel free to modify it or add your own variations to make it more interesting.

LLaMA-3.1-8B-TAPO:

Here's a folk-style melody in ABC notation:

X:1 T: The Wanderer's Lament **M**: 4/4 **L**: 1/4=120 **K**: C

C4 E4 G4 C4 | E4 G4 A4 G4 | F4 E4 D4 C4 | E4 G4 A4 G4 |

C4 E4 G4 C4 | A4 G4 F4 E4 | D4 C4 D4 E4 | F4 E4 D4 C4 |

G4 A4 G4 F4 | E4 D4 C4 E4 | G4 F4 E4 D4 | C4 E4 G4 C4 |

This melody is written in the key of C major, with a moderate tempo and a simple, repetitive structure. The ABC notation uses the following symbols:

- `X:1` indicates that this is the first tune in the file.
- `T:` is the title of the tune.
- `M:` specifies the meter (4/4 in this case).
- `L:` specifies the tempo (120 quarter notes per minute).
- `K:` specifies the key (C major).
- The lines of notes are written in a simple, letter-based notation, with `C4` indicating a C note an octave above middle C, and `E4` indicating an E note an octave above middle C, and so on.

I hope you enjoy this folk-style melody!

Figure 3: Comparison between *LLaMA-3.1-8B-Instruct* and *LLaMA-3.1-8B-TAPO* from three cases in Arena-Hard.

the Kanji character 'Wish' by providing multiple thoughtfully separated options, each clearly formatted and logically structured. This enhances clarity and allows learners to identify and select a mnemonic that resonates with them, compared to the single, less engaging option by *LLaMA-3.1-8B-Instruct*. In the second case (center of Figure 3), *LLaMA-3.1-8B-TAPO* improves readability and comprehension by using bold formatting to emphasize each occurrence of 'bug' and aligning explanations with a numbered list. This ensures better organization and enables readers to quickly grasp the context and meaning of each instance, whereas *LLaMA-3.1-8B-Instruct*'s less structured formatting is harder to follow. In the third case (bottom of Figure 3), *LLaMA-3.1-8B-TAPO* organizes a folk-style melody with logical grouping of notes and appropriate line breaks, enhancing readability and usability. In contrast, the folk-style melody output by *LLaMA-3.1-8B-Instruct* suffers from fragmented line breaks, splitting logical sequences of notes into disjointed segments, which disrupts the logical flow and makes it challenging to interpret and perform the melody accurately.

8 Conclusion

In this paper, we conducted the first exploration of textual aesthetics in LLMs and introduced a series of techniques to enhance the aesthetic quality of LLMs outputs. First, we developed the TEXAES dataset, the first textual aesthetic dataset in the LLMs domain, using our specially-designed data polishing pipeline. Based on this dataset, we proposed the TAPO, which fine-tunes LLMs to improve the aesthetic quality of their outputs while preserving their core capabilities. Both qualitative and quantitative experiments validated the effectiveness of our proposed techniques. We hope our work serves as an early exploration of textual aesthetics in LLMs and provides valuable support for researchers in the open-source community. In future work, we will continue to explore ways to collect diverse and high-quality textual aesthetics data, while designing more efficient and effective tuning techniques for aesthetic fine-tuning.

Limitations

The TEXAES dataset was constructed from UltraFeedback. While UltraFeedback is a large-scale and diverse dataset, it still has certain limitations. This reliance may constrain the diversity of textual styles, genres, and languages represented. Expanding the dataset to cover a broader range of languages, writing styles, and specialized domains could enhance its generalization capability. Another limitation is that we employed LLMs as judges to simulate human annotators' preferences for textual aesthetics. Although this approach has been adopted in recent research and has demonstrated high agreement with human evaluations (Li et al., 2023; Dubois et al., 2024; Zheng et al., 2023; Li et al., 2024), LLMs still cannot precisely model human textual aesthetic preferences in all contexts. In the future, we plan to explore more generalized and accurate methods for assessing the textual aesthetics of LLMs, such as developing an Ensemble-as-Judges system, similar to Judges with Jury (Verga et al., 2024), to reduce biases and improve correlation.

Ethics Statement

We used the dataset UltraFeedback, which is released under the MIT License, allowing free use, copy, modify, merge, and publish. We release our dataset under the same license. Building on this foundation, this paper presents work that aims to advance the field of Natural Language Processing, specifically Large Language Models. There are potential societal consequences of our work associated with LLMs, such as AI safety and reliability. Beyond LLMs, we believe no other consequences need to be highlighted here.

References

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Subhabrata Bhattacharya, Rahul Sukthankar, and Mubarak Shah. 2010. A framework for photo-quality assessment and enhancement based on visual aesthetics. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 271–280.

Sebastian Bosse, Dominique Maniry, Klaus-Robert Müller, Thomas Wiegand, and Wojciech Samek. 2017. Deep neural networks for no-reference and full-reference image quality assessment. *IEEE Transactions on image processing*, 27(1):206–219.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. 2024. Ultrafeedback: Boosting language models with scaled ai feedback. In *Forty-first International Conference on Machine Learning*.

Yubin Deng, Chen Change Loy, and Xiaoou Tang. 2017. Image aesthetic assessment: An experimental survey. *IEEE Signal Processing Magazine*, 34(4):80–106.

Yubin Deng, Chen Change Loy, and Xiaoou Tang. 2018. Aesthetic-driven image enhancement by adversarial learning. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 870–878.

William DuBay. 2004. The principles of readability. *Impact Information*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, et al. 2024. The llama 3 herd of models.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Johannes Fürnkranz and Eyke Hüllermeier. 2010. Preference learning and ranking by pairwise comparison. In *Preference learning*, pages 65–82. Springer.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. 2022. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Yipo Huang, Xiangfei Sheng, Zhichao Yang, Quan Yuan, Zhichao Duan, Pengfei Chen, Leida Li, Weisi Lin, and Guangming Shi. 2024a. Aesexpert: Towards multi-modality foundation model for image aesthetics perception. *arXiv preprint arXiv:2404.09624*.

Yipo Huang, Quan Yuan, Xiangfei Sheng, Zhichao Yang, Haoning Wu, Pengfei Chen, Yuzhe Yang, Leida Li, and Weisi Lin. 2024b. Aesbench: An expert benchmark for multimodal large language models on image aesthetics perception. *arXiv preprint arXiv:2401.08276*.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. 2021. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5148–5157.

Shu Kong, Xiaohui Shen, Zhe Lin, Radomir Mech, and Charless Fowlkes. 2016. Photo aesthetics ranking network with attributes and content adaptation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 662–679. Springer.

Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018. Can neural machine translation be improved with user feedback? *arXiv preprint arXiv:1804.05958*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Dingquan Li, Tingting Jiang, Weisi Lin, and Ming Jiang. 2018. Which has better visual quality: The clear blue sky or a blurry animal? *IEEE Transactions on Multimedia*, 21(5):1221–1234.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.

R Duncan Luce. 1959. *Individual choice behavior*, volume 4. Wiley New York.

Naila Murray, Luca Marchesotti, and Florent Perronnin. 2012. Ava: A large-scale database for aesthetic visual analysis. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2408–2415. IEEE.

Frank Nack, Chitra Dorai, and Svetha Venkatesh. 2001. Computational media aesthetics: Finding meaning beautiful. *IEEE multimedia*, 8(4):10–12.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

L Neumann, M Sbert, B Gooch, W Purgathofer, et al. 2005. Defining computational aesthetics. *Computational aesthetics in graphics, visualization and imaging*, 2005:13–18.

OpenAI. 2023. Introducing chatgpt.

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

Robin L Plackett. 1975. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202.

Guanyi Qin, Runze Hu, Yutao Liu, Xiawu Zheng, Haotian Liu, Xiu Li, and Yan Zhang. 2023. Data-efficient image quality assessment with attention-panel decoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2091–2100.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024a. Direct preference optimization: Your language model is secretly a reward model.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024b. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

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

Charlie Snell, Ilya Kostrikov, Yi Su, Mengjiao Yang, and Sergey Levine. 2022. Offline rl for natural language generation with implicit language q learning. *arXiv preprint arXiv:2206.11871*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Hsiao-Hang Su, Tse-Wei Chen, Chieh-Chi Kao, Winston H Hsu, and Shao-Yi Chien. 2011. Scenic photo quality assessment with bag of aesthetics-preserving features. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1213–1216.

Shaolin Su, Qingsen Yan, Yu Zhu, Cheng Zhang, Xin Ge, Jinqiu Sun, and Yanning Zhang. 2020. Blindly assess image quality in the wild guided by a self-adaptive hyper network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3667–3676.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023a. Llama 2: Open foundation and fine-tuned chat models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Gerdineke Van Silfhout, Jacqueline Evers-Vermeul, Willem M Mak, and Ted JM Sanders. 2014. Connectives and layout as processing signals: How textual features affect students' processing and text representation. *Journal of Educational Psychology*, 106(4):1036.

Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Haoning Wu, Zicheng Zhang, Erli Zhang, Chaofeng Chen, Liang Liao, Annan Wang, Kaixin Xu, Chunyi Li, Jingwen Hou, Guangtao Zhai, et al. 2024a. Q-instruct: Improving low-level visual abilities for multi-modality foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25490–25500.

Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. 2023. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090*.

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*.

Xun Wu, Shaohan Huang, and Furu Wei. 2024b. Multimodal large language model is a human-aligned annotator for text-to-image generation. *arXiv preprint arXiv:2404.15100*.

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

Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. 2022. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1191–1200.

Shuai Zhang and Yutao Liu. 2023. Multi-scale transformer with decoder for image quality assessment. In *CAAI International Conference on Artificial Intelligence*, pages 220–231. Springer.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv preprint arXiv:2403.13372*.

A Dataset Statistics

Figure 4 illustrates the difference in token length between the text that has undergone aesthetic polishing and the original text. The mean length difference is 49 tokens, with the 25th and 75th percentile values being -7 and 54, respectively. The maximum length difference is 2673 tokens, while the minimum length difference is -1024 tokens.

In Figure 5, we present the length distribution of TEXAES and UltraFeedback (Cui et al., 2024). For TEXAES, the mean length is 293 tokens, with the 25th and 75th percentiles at 97 and 444, respectively, and a maximum length of 1408 tokens. For UltraFeedback, the mean length is 297 tokens, with the 25th and 75th percentiles at 77 and 464, and a maximum length of 2700 tokens.

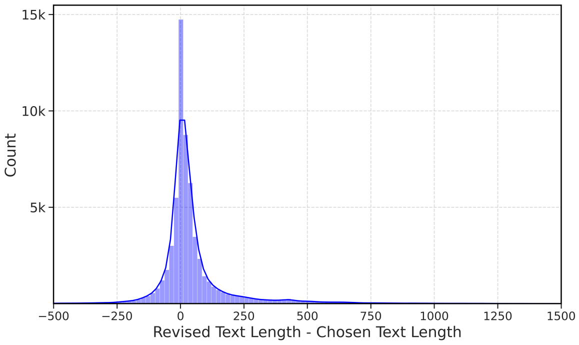


Figure 4: Distribution of length differences.

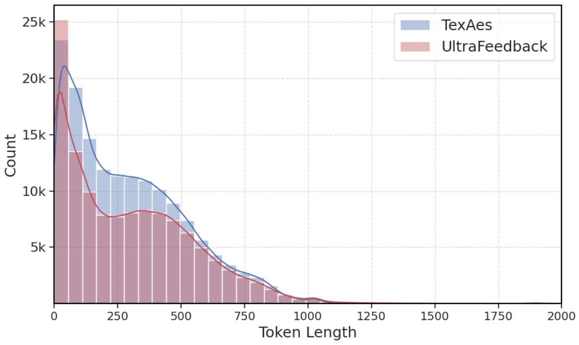


Figure 5: Comparison of token length distributions between TEXAES and UltraFeedback.

B Training Parameters

We present the details of the experimental settings in Table 5 and Table 6. For the sake of fairness in comparison, we used the same training parameters as those employed by DPO during the preference optimization stage. Our experiments are based on Llama-Factory (Zheng et al., 2024)

C Mathematical Derivations

In this section, we prove that $\mathcal{L}_{\text{DPO}}$ from Eq. 3 is equivalent to Eq. 1. To begin, consider Eq.3:

$$\begin{aligned}\mathcal{L}_{\text{DPO}} &= -\log\left(\frac{\exp(r_\theta(x,y_w))}{\sum_{i\in\{w,l\}}\exp(r_\theta(x,y_i))}\right)\\ &= -\log\left(\frac{\exp(r_\theta(x,y_w))}{\exp(r_\theta(x,y_w))+\exp(r_\theta(x,y_l))}\right)\\ &= -\log\left(\frac{1}{1+\exp(r_\theta(x,y_l)-r_\theta(x,y_w))}\right)\\ &= -\log\sigma\left(r_\theta(x,y_w)-r_\theta(x,y_l)\right)\end{aligned} \tag{5}$$

Here, σ denotes the sigmoid function. In Section 4.2, we presented the specific expressions for $r_\theta(x, y_w)$ and $r_\theta(x, y_l)$:

$$r_\theta(x,y_w) = \beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)}, \quad r_\theta(x,y_l) = \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \tag{6}$$

By substituting Eq. 6 into Eq. 5, we obtain:

$$\mathcal{L}_{\text{DPO}} = -\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right) \tag{7}$$

This shows that $\mathcal{L}_{\text{DPO}}$ as defined in Eq. 3 is indeed equivalent to Eq. 1, thus completing the proof.

Parameter	**Value**
Training Method	LoRA (Hu et al., 2021)
Maximum Sequence Length	2048
Optimizer	AdamW
Precision	BFloat16
Global Batch Size	64
Maximum Learning Rate	0.0002
Learning Rate Scheduler	Cosine with 10% Warmup
Number of Epochs	2

Table 5: Parameters for SFT training.

Parameter	**Value**
Training Method	LoRA (Hu et al., 2021)
Maximum Sequence Length	2048
Optimizer	AdamW
Precision	BFloat16
Global Batch Size	64
Maximum Learning Rate	0.00002
Learning Rate Scheduler	Cosine with 10% Warmup
Number of Epochs	2
Beta	0.1
Loss Weight w_{TA}	1
Loss Weight w_{DPO}	1

Table 6: Parameters for TAPO training.

D Additional Experiment Results

D.1 Length Constraint in TexAes Dataset

To verify whether filtering out outliers in the distribution of length differences before and after aesthetic polishing can improve the quality of TexAes during its construction phase, we conducted an ablation experiment on data without length filtering. Specifically, the model was trained based on *LLaMA-3.1-8B-Base* using DPO with y_t as the chosen response and y_l as the rejected response, and the results are presented in Table 7. The findings demonstrate that the performance of the model, after removing data points with excessive length deviations, significantly exceeds that of the model trained without such length filtering across all evaluation tasks. Furthermore, a statistical analysis of the output lengths generated by the model revealed that the outputs produced by the model trained with length-filtered data were not only shorter but also more concise, thereby affirming the efficacy of length filtering in text aesthetic optimization.

Length Filter	TA Text	TA Image	AlpacaEval 2.0	Arena-Hard	MT-Bench	Avg Tokens
✘	24.94	56.64	20.62	7.57	4.75	649
✓	**25.79**	**60.64**	**24.06**	**9.04**	**5.78**	610

Table 7: Results of the ablation study on length filtering: comparison of model performance across various evaluation metrics with and without length filtering.

D.2 Generalizability of TEXAES and TAPO to Other LLMs

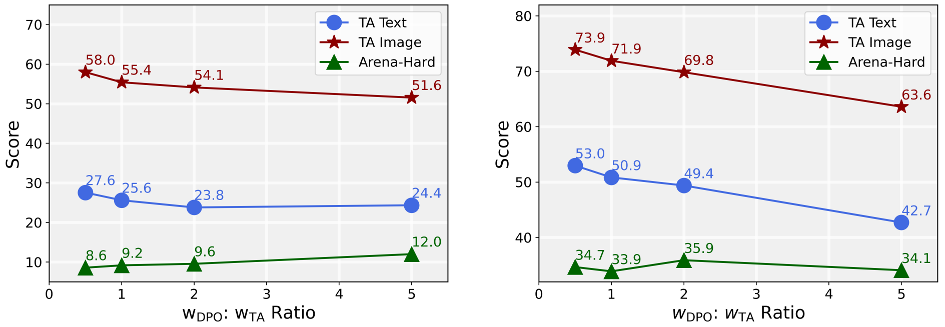
To evaluate the generalizability of the proposed TEXAES dataset and the TAPO method beyond the LLaMA series, we conducted additional experiments on two other widely used large language models: Qwen2-7B-Instruct (Yang et al., 2024) and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023).

In these experiments, TEXAES was used as the training dataset, and TAPO was applied as the training method under the same experimental settings as those used for LLaMA-3.1-8B-Instruct. We conducted comparative experiments using DPO training, with y_t and y_w as the chosen responses and y_l as the rejected response, denoted as DPO(y_t, y_l) and DPO(y_w, y_l), respectively. Here y_t, y_w, and y_l are responses in the TEXAES dataset. The evaluation results, summarized in Table 8, indicate that although direct training with DPO(y_t, y_l) can improve textual aesthetics scores, it leads to a decline in general performance (e.g., Qwen2-7B-Instruct showed decreased performance on Arena-Hard, MT-Bench, MMLU (Hendrycks et al., 2020), and Mistral-7B-Instruct-v0.3 showed decreased performance on MT-Bench, MMLU). Using DPO(y_w, y_l) can enhance model performance on most general tasks, except for a slight decline for Qwen2-7B-Instruct on MT-Bench. However, it does not improve performance in aesthetic evaluation tasks. In contrast, both Qwen2-7B-Instruct and Mistral-7B-Instruct-v0.3, following the application of TAPO, demonstrated significant improvements across all benchmarks. Additionally, models trained with TAPO surpassed those trained with DPO(y_t, y_l) and DPO(y_w, y_l) in both textual aesthetics and general response capabilities.

These findings are consistent with the results observed in the LLaMA-3.1 series, providing compelling evidence of the broad applicability and effectiveness of the TEXAES dataset and the TAPO method across diverse LLM architectures.

Model	TA Text WR(%)	TA Image WR(%)	AlpacaEval 2.0 LC WR(%)	Arena-Hard WR(%)	MT-Bench Avg. Score	MMLU 5-shot
Qwen2-7B-Instruct						
Qwen2-7B-Instruct (Yang et al., 2024)	24.63	39.40	33.43	27.69	7.48	70.46
Qwen2-7B-Instruct + DPO (y_w, y_l)	15.45	37.46	36.25	29.68	7.38	**70.52**
Qwen2-7B-Instruct + DPO (y_t, y_l)	33.84	61.23	40.16	25.30	7.19	70.34
Qwen2-7B-Instruct-TAPO	**37.99**	**64.28**	**40.27**	**32.40**	**7.48**	70.49
Mistral-7B-Instruct-v0.3						
Mistral-7B-Instruct-v0.3 (Jiang et al., 2023)	8.26	28.90	29.87	17.13	6.59	61.52
Mistral-7B-Instruct-v0.3 + DPO (y_w, y_l)	8.17	30.84	31.07	18.79	6.74	**61.68**
Mistral-7B-Instruct-v0.3 + DPO (y_t, y_l)	25.59	54.64	36.78	20.83	6.56	61.36
Mistral-7B-Instruct-v0.3-TAPO	**28.55**	**57.84**	**38.53**	**23.10**	**6.80**	61.55

Table 8: Performance comparison of Qwen2-7B-Instruct and Mistral-7B-Instruct-v0.3 models after training with TEXAES and TAPO. DPO(y_w, y_l) denotes DPO training where y_w is the chosen response and y_l is the rejected response. DPO(y_t, y_l) indicates DPO training where y_t is the chosen response and y_l is the rejected response. Here, y_t, y_w, and y_l are responses in the TexAes dataset.



(a) LLaMA-3.1-8B-Base

(b) LLaMA-3.1-8B-Instruct

Figure 6: Performance Across Various Weight Ratios

D.3 Impact of Loss Weight

To determine the influence of the weight ratio between $\mathcal{L}_{\text{TA}}$ and $\mathcal{L}_{\text{DPO}}$ in TAPO on the aesthetics of the texts of the model and the overall performance, we performed a series of methodical experiments. Specifically, we experimented with two settings: 1. First, we used the Tulu-v2 dataset (Ivison et al., 2023) to fine-tune the *LLaMA-3.1-8B-base* model in a supervised manner, followed by further optimization using TAPO; 2. Second, we directly applied TAPO to the *LLaMA-3.1-8B-instruct* model. We set the weight ratios of $\mathcal{L}_{\text{TA}}$ to $\mathcal{L}_{\text{DPO}}$ at 2:1, 1:1, 1:2 and 1:5, respectively, to train the models. We then evaluated the models' text-based and image-based text aesthetic scores, as well as their performance on Arena-Hard.

Figures 6a and 6b illustrate the performance variations of TAPO across different weight ratios. For the *LLaMA-3.1-8B-base* model, increasing the proportion of $\mathcal{L}_{\text{DPO}}$ consistently improves the Arena-Hard score but decreases both text-based and image-based text aesthetic scores. This indicates that a higher proportion of $\mathcal{L}_{\text{DPO}}$ improves optimization toward human preference at the expense of aesthetic preference. For the *LLaMA-3.1-8B-instruct* model, which is already aligned with human preferences, further increasing $\mathcal{L}_{\text{DPO}}$ yields limited improvements in instruct-following capability and significantly decreases textual aesthetic preference.

D.4 TEXAES vs. UltraFeedback

To validate the effectiveness of the TEXAES data set, we performed a comparative analysis of models trained using TEXAES against those trained with UltraFeedback data. We applied the Direct Preference Optimization (DPO) method to align human preferences with the y_w and y_l pairs from UltraFeedback and the y_t and y_l pairs from TEXAES. The experiments were conducted on both the *LLaMA-3.1-Base* and *LLaMA3.1-Instruct* models.

Dataset	TA Text WR(%)	TA Image WR(%)	AlpacaEval 2.0 LC WR(%)	Arena-Hard WR(%)	MT-Bench Avg. Score	MMLU 5-shot
LLaMA-3.1-8B-Base						
Baseline	1.17	8.60	5.24	4.10	5.60	64.07
UltraFeedback	2.56	8.17	9.29	7.06	**5.92**	**65.02**
TEXAES	**25.79**	**60.64**	**24.06**	**9.04**	5.78	63.17
LLaMA-3.1-8B-Instruct						
Baseline	33.42	47.94	41.34	**37.10**	7.42	68.80
UltraFeedback	30.92	48.57	44.19	34.74	**7.76**	**68.90**
TEXAES	**49.07**	**68.63**	**45.82**	29.87	7.55	68.52

Table 9: Comparative analysis of TEXAES and UltraFeedback with DPO Training. The baseline represents the performance of LLaMA-3.1-8B-Base which is fine-tuned using the Tulu-v2 dataset and LLaMA-3.1-8B-Instruct.

The results, shown in Table 9, indicate that for the *LLaMA-3.1-Base* model, UltraFeedback improved performance in AlpacaEval 2.0, Arena-Hard, MT-Bench and MMLU (Hendrycks et al., 2020). For the *LLaMA3.1-Instruct* model, there were performance improvements across most tasks, except for a slight decline in Arena-Hard. However, UltraFeedback did not improve performance in aesthetic evaluation tasks. Models trained with TEXAES showed significant performance improvements over those trained with UltraFeedback in most tasks on the *LLaMA-3.1-Base* model, with a minor decrease in MMLU. For the *LLaMA3.1-Instruct* model, the one trained with TEXAES exhibited general capabilities comparable to those of the UltraFeedback-trained model while surpassing it in aesthetic tasks. These experiments demonstrate that TEXAES not only optimizes the textual aesthetic performance of large language models but also aligns well with human preferences.

D.5 Criteria for Reject Sample Selection

To effectively optimize textual aesthetics using preference optimization, it is essential to construct preference pairs consisting of chosen and rejected responses. For our purposes, we select y_t from TEXAES as the chosen response. As the rejected response, we use either the original chosen response y_w or the original rejected response y_l from the UltraFeedback dataset. We conducted DPO experiments to compare the impact of y_w and y_l on the model's performance. The results are presented in Table 10.

Training Settings	**TA Text WR(%)**	**TA Image WR(%)**	**AlpacaEval 2.0 LC WR(%)**	**Arena-Hard WR(%)**	**MT-Bench Avg. Score**	**MMLU 5-shot**
LLaMA-3.1-8B-Base						
Baseline	1.17	8.60	5.24	4.10	5.60	**64.07**
DPO(y_t, y_w)	15.72	51.70	15.78	4.10	5.19	50.36
DPO(y_t, y_l)	**25.79**	**60.64**	**24.06**	**9.04**	**5.78**	63.17
LLaMA-3.1-8B-Instruct						
Baseline	33.42	47.94	41.34	**37.10**	7.42	**68.80**
DPO(y_t, y_w)	46.89	**71.19**	38.93	26.04	7.36	68.31
DPO(y_t, y_l)	**49.07**	68.63	**45.82**	29.87	**7.55**	68.52

Table 10: Evaluation of performance across different rejected samples. DPO(y_w, y_l) denotes DPO training where y_w is the chosen response and y_l is the rejected response. Similarly, DPO(y_t, y_l) refers to DPO training where y_t is the chosen response and y_l is the rejected response. Here, y_t, y_w, and y_l are responses in the TEXAES dataset.

The model trained with DPO(y_t, y_l) outperforms the model trained with DPO(y_t, y_w) in most tasks. This superior performance is attributed to DPO(y_t, y_l) optimizing both textual aesthetic preferences and human preferences. When using y_w as the rejected response, the model trained with DPO(y_t, y_w) significantly outperforms the baseline model in text aesthetic scores but shows a decrease in performance in other tasks. This indicates that y_t is more aligned with aesthetic preferences, meeting our expectations.

D.6 Detailed Analysis of Aesthetics-Driven Improvements Across Four Dimensions

Model	**Clarity**	**Layout**	**Consistency**	**Coherence**	**Average**
Mistral-7B-Instruct-v0.3					
Baseline	24.29	27.62	42.78	18.11	28.20
TAPO	**32.15**	**66.76**	**47.58**	**32.55**	**44.76**
Qwen2-7B-Instruct					
Baseline	33.48	39.48	45.74	32.05	37.69
TAPO	**37.31**	**68.46**	**48.88**	**39.26**	**48.48**
LLaMA-3.1-8B-Instruct					
Baseline	32.74	55.96	44.44	32.66	41.45
TAPO	**42.26**	**80.40**	**55.49**	**42.11**	**55.07**
Meta-Llama-3.1-70B-Instruct					
Baseline	46.42	68.77	51.43	46.16	53.20
TAPO	**50.94**	**84.69**	**58.43**	**54.55**	**62.15**

Table 11: Comparison of baseline and TAPO models across four dimensions of textual aesthetics.

To further substantiate the relationship between textual aesthetics and improvements in text consistency and coherence, we conducted a detailed evaluation using GPT-4o as an annotator across four dimensions: *Clarity*, *Layout*, *Consistency*, and *Coherence*. We followed the **Text-Based Text Aesthetics Scoring** method, where each model π_i was compared with a strong baseline π_{base} (GPT-4-0314) across the four dimensions of textual aesthetics using a 5-point Likert scale (Likert, 1932). Win rates for each dimension were then derived using the Bradley-Terry model (Bradley and Terry, 1952). The judge prompts are provided in Appendix G.4, and a detailed example is included in Appendix F.4.. The results, as shown in Table 11, indicate that TAPO-enhanced models outperform their baselines not only in overall aesthetics, but also in consistency and coherence. These findings are further supported by case studies (Figure 3), where TAPO-trained models produce outputs that are structurally clearer and semantically more cohesive than their baseline counterparts. This empirical evidence demonstrates that enhancing textual aesthetics through our method has measurable and interpretable downstream benefits for both coherence and consistency.

E Human Annotation Details

In our study, we employed three annotators: two graduate students in computer science and one professor with a background in applied linguistics. All three evaluators are non-native English speakers but are proficient in English. Their diverse academic and linguistic backgrounds provide a balanced perspective for assessing textual aesthetics across the four key dimensions—clarity, layout, uniformity, and coherence.

The annotators underwent a comprehensive training and calibration process prior to the main evaluation. This training ensured that their understanding of the evaluation criteria was consistent and aligned. Annotators were introduced to the four evaluation dimensions—clarity (ease of comprehension), layout (visual organization), uniformity (consistent formatting), and coherence (logical structure)—with detailed explanations and examples. They practiced with a subset of the dataset, and their evaluations were reviewed with feedback provided to refine their approach. A final readiness test was conducted to confirm alignment and preparedness for the main evaluation phase.

For the evaluation, 50 prompts were randomly selected from the Arena-Hard (Li et al., 2024), and all models under evaluation were tasked with generating responses to these prompts using identical parameters. This ensured consistency in the generation process and a fair basis for comparison across models. The generated text samples were anonymized and presented in a standardized format, removing all identifying information about the originating model or source.

The annotators independently evaluated these samples without communication or influence from others, maintaining impartiality throughout the process. A pairwise comparison methodology was employed, where annotators assigned scores in the form of win, tie, or loss for each sample comparison across the four evaluation dimensions.

E.1 Computational Overhead of the Aesthetics Optimization Process

To further address concerns regarding the potential computational overhead introduced by aesthetics optimization, we analyzed both training and inference costs under consistent experimental conditions.

Training Overhead. Under the same settings as in Table 6, where training was conducted using 8 NVIDIA A100 GPUs (40 GB VRAM), DPO training for the Meta-Llama-3.1-8B-Instruct model required approximately 3.8 hours, while TAPO training took around 4.2 hours. The additional time is minimal, suggesting that TAPO introduces only a slight increase in training time relative to DPO.

Model	Average Length
Meta-Llama-3.1-8B-Instruct	
Meta-Llama-3.1-8B-Instruct	731
Meta-Llama-3.1-8B-Instruct-DPO	631
Meta-Llama-3.1-8B-Instruct-TAPO	623
Mistral-7B-Instruct-v0.3	
Mistral-7B-Instruct-v0.3	482
Mistral-7B-Instruct-v0.3-DPO	570
Mistral-7B-Instruct-v0.3-TAPO	583
Qwen2-7B-Instruct	
Qwen2-7B-Instruct	579
Qwen2-7B-Instruct-DPO	583
Qwen2-7B-Instruct-TAPO	589

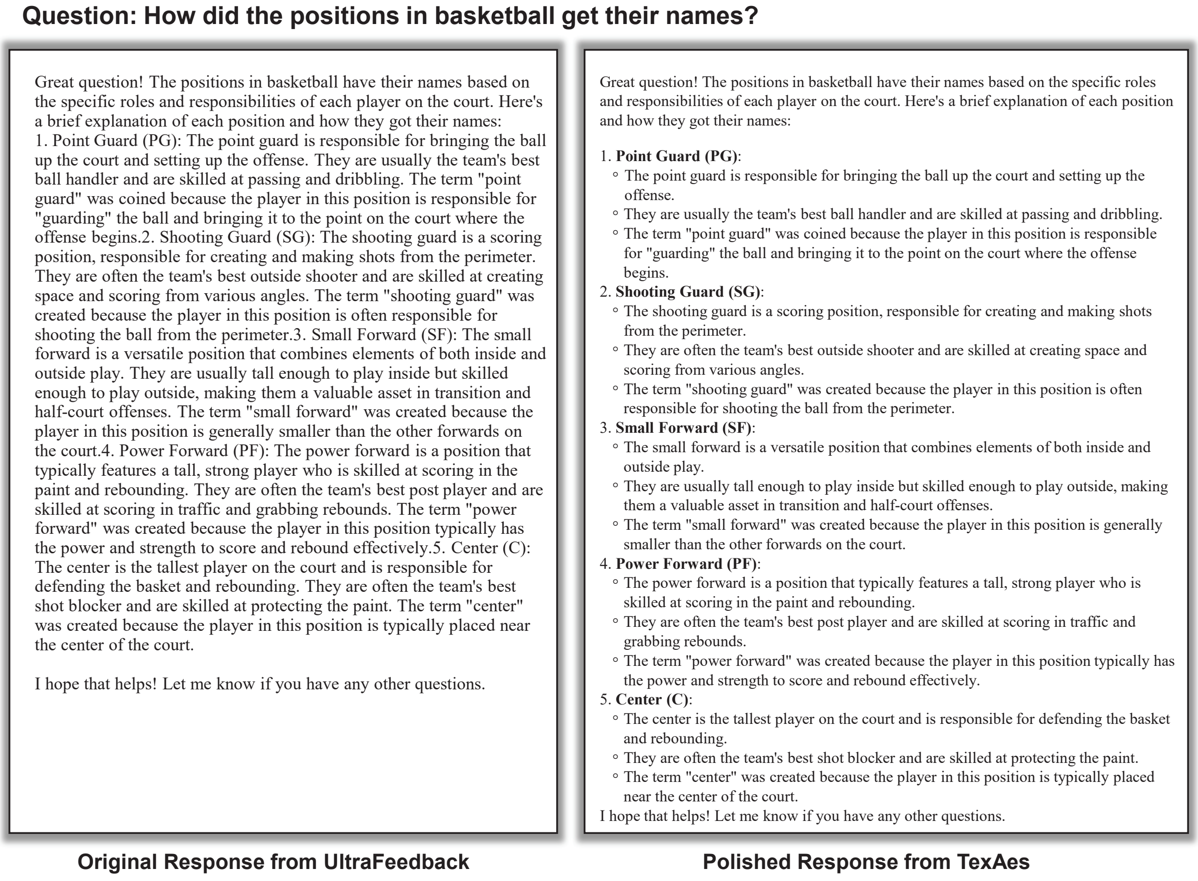
Table 12: Average Token Lengths

Inference Overhead. We also measured inference costs by comparing output token lengths on the Arena-Hard dataset, using UltraFeedback for DPO training and TEXAES for TAPO training. As reported in Table 12, both DPO and TAPO optimized models produce outputs with average lengths similar to their respective original models. The differences are small and not statistically significant, indicating that TAPO does not substantially increase inference costs.

Summary. TAPO introduces a modest increase in training time but does not significantly affect inference efficiency. These results confirm that the computational overhead of aesthetics optimization is negligible, while preserving the performance benefits of TAPO.

F Additional Examples

F.1 Text Aesthetic Polishing Case



Question: How did the positions in basketball get their names?

Great question! The positions in basketball have their names based on the specific roles and responsibilities of each player on the court. Here's a brief explanation of each position and how they got their names:
1. Point Guard (PG): The point guard is responsible for bringing the ball up the court and setting up the offense. They are usually the team's best ball handler and are skilled at passing and dribbling. The term "point guard" was coined because the player in this position is responsible for "guarding" the ball and bringing it to the point on the court where the offense begins.2. Shooting Guard (SG): The shooting guard is a scoring position, responsible for creating and making shots from the perimeter. They are often the team's best outside shooter and are skilled at creating space and scoring from various angles. The term "shooting guard" was created because the player in this position is often responsible for shooting the ball from the perimeter.3. Small Forward (SF): The small forward is a versatile position that combines elements of both inside and outside play. They are usually tall enough to play inside but skilled enough to play outside, making them a valuable asset in transition and half-court offenses. The term "small forward" was created because the player in this position is generally smaller than the other forwards on the court.4. Power Forward (PF): The power forward is a position that typically features a tall, strong player who is skilled at scoring in the paint and rebounding. They are often the team's best post player and are skilled at scoring in traffic and grabbing rebounds. The term "power forward" was created because the player in this position typically has the power and strength to score and rebound effectively.5. Center (C): The center is the tallest player on the court and is responsible for defending the basket and rebounding. They are often the team's best shot blocker and are skilled at protecting the paint. The term "center" was created because the player in this position is typically placed near the center of the court.

I hope that helps! Let me know if you have any other questions.

Original Response from UltraFeedback

Great question! The positions in basketball have their names based on the specific roles and responsibilities of each player on the court. Here's a brief explanation of each position and how they got their names:

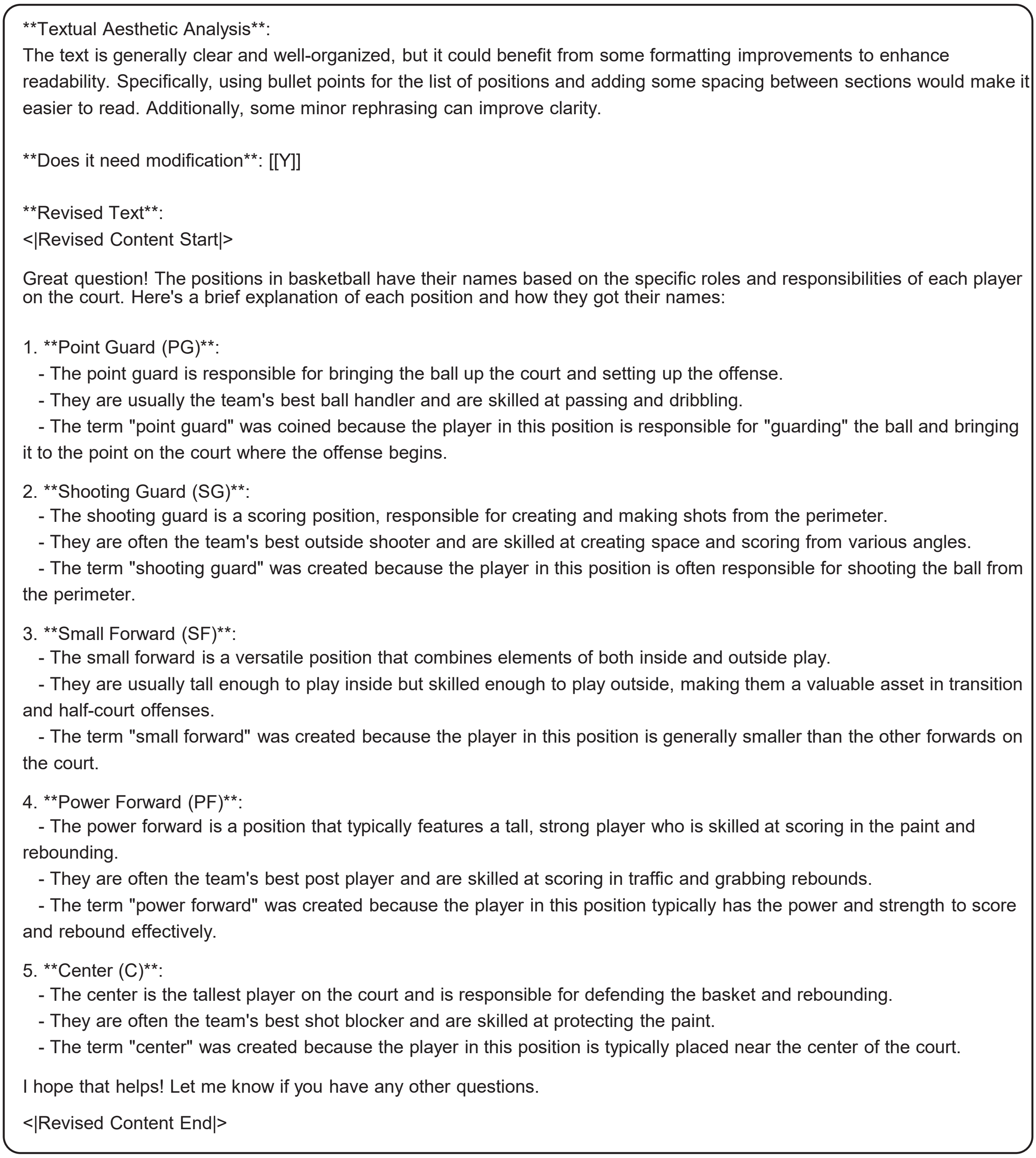
1. **Point Guard (PG)**:
   - ◦ The point guard is responsible for bringing the ball up the court and setting up the offense.
   - ◦ They are usually the team's best ball handler and are skilled at passing and dribbling.
   - ◦ The term "point guard" was coined because the player in this position is responsible for "guarding" the ball and bringing it to the point on the court where the offense begins.
2. **Shooting Guard (SG)**:
   - ◦ The shooting guard is a scoring position, responsible for creating and making shots from the perimeter.
   - ◦ They are often the team's best outside shooter and are skilled at creating space and scoring from various angles.
   - ◦ The term "shooting guard" was created because the player in this position is often responsible for shooting the ball from the perimeter.
3. **Small Forward (SF)**:
   - ◦ The small forward is a versatile position that combines elements of both inside and outside play.
   - ◦ They are usually tall enough to play inside but skilled enough to play outside, making them a valuable asset in transition and half-court offenses.
   - ◦ The term "small forward" was created because the player in this position is generally smaller than the other forwards on the court.
4. **Power Forward (PF)**:
   - ◦ The power forward is a position that typically features a tall, strong player who is skilled at scoring in the paint and rebounding.
   - ◦ They are often the team's best post player and are skilled at scoring in traffic and grabbing rebounds.
   - ◦ The term "power forward" was created because the player in this position typically has the power and strength to score and rebound effectively.
5. **Center (C)**:
   - ◦ The center is the tallest player on the court and is responsible for defending the basket and rebounding.
   - ◦ They are often the team's best shot blocker and are skilled at protecting the paint.
   - ◦ The term "center" was created because the player in this position is typically placed near the center of the court.

I hope that helps! Let me know if you have any other questions.

Polished Response from TexAes

Figure 7: An additional comparison between UltraFeedback and TEXAES. The polished response offers a structured explanation of basketball positions that is easier to read and understand compared to the original response from UltraFeedback.

F.2 Text Aesthetic Polishing Process



Textual Aesthetic Analysis:
The text is generally clear and well-organized, but it could benefit from some formatting improvements to enhance readability. Specifically, using bullet points for the list of positions and adding some spacing between sections would make it easier to read. Additionally, some minor rephrasing can improve clarity.

Does it need modification: [[Y]]

Revised Text:
<|Revised Content Start|>

Great question! The positions in basketball have their names based on the specific roles and responsibilities of each player on the court. Here's a brief explanation of each position and how they got their names:

1. **Point Guard (PG)**:
   - The point guard is responsible for bringing the ball up the court and setting up the offense.
   - They are usually the team's best ball handler and are skilled at passing and dribbling.
   - The term "point guard" was coined because the player in this position is responsible for "guarding" the ball and bringing it to the point on the court where the offense begins.

2. **Shooting Guard (SG)**:
   - The shooting guard is a scoring position, responsible for creating and making shots from the perimeter.
   - They are often the team's best outside shooter and are skilled at creating space and scoring from various angles.
   - The term "shooting guard" was created because the player in this position is often responsible for shooting the ball from the perimeter.

3. **Small Forward (SF)**:
   - The small forward is a versatile position that combines elements of both inside and outside play.
   - They are usually tall enough to play inside but skilled enough to play outside, making them a valuable asset in transition and half-court offenses.
   - The term "small forward" was created because the player in this position is generally smaller than the other forwards on the court.

4. **Power Forward (PF)**:
   - The power forward is a position that typically features a tall, strong player who is skilled at scoring in the paint and rebounding.
   - They are often the team's best post player and are skilled at scoring in traffic and grabbing rebounds.
   - The term "power forward" was created because the player in this position typically has the power and strength to score and rebound effectively.

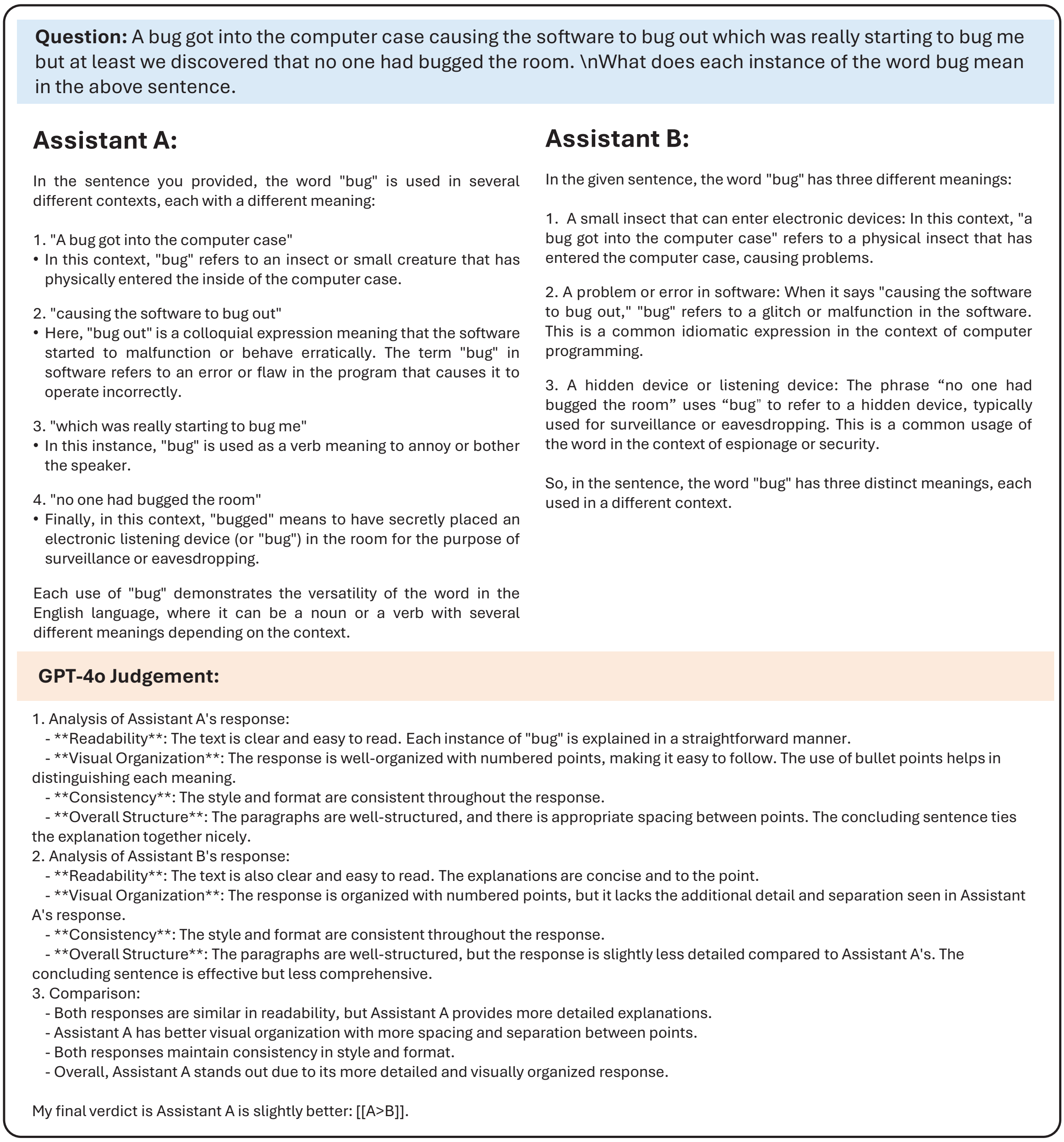
5. **Center (C)**:
   - The center is the tallest player on the court and is responsible for defending the basket and rebounding.
   - They are often the team's best shot blocker and are skilled at protecting the paint.
   - The term "center" was created because the player in this position is typically placed near the center of the court.

I hope that helps! Let me know if you have any other questions.

<|Revised Content End|>

Figure 8: Textual Aesthetic Polishing by GPT-4o: This case illustrates the process where GPT-4o performs semantic analysis and aesthetic evaluation, identifies the need for modification, and offers a revised version of the text.

F.3 Text aesthetics judgement



Question: A bug got into the computer case causing the software to bug out which was really starting to bug me but at least we discovered that no one had bugged the room. \nWhat does each instance of the word bug mean in the above sentence.

Assistant A:

In the sentence you provided, the word "bug" is used in several different contexts, each with a different meaning:

1. "A bug got into the computer case"
• In this context, "bug" refers to an insect or small creature that has physically entered the inside of the computer case.

2. "causing the software to bug out"
• Here, "bug out" is a colloquial expression meaning that the software started to malfunction or behave erratically. The term "bug" in software refers to an error or flaw in the program that causes it to operate incorrectly.

3. "which was really starting to bug me"
• In this instance, "bug" is used as a verb meaning to annoy or bother the speaker.

4. "no one had bugged the room"
• Finally, in this context, "bugged" means to have secretly placed an electronic listening device (or "bug") in the room for the purpose of surveillance or eavesdropping.

Each use of "bug" demonstrates the versatility of the word in the English language, where it can be a noun or a verb with several different meanings depending on the context.

Assistant B:

In the given sentence, the word "bug" has three different meanings:

1. A small insect that can enter electronic devices: In this context, "a bug got into the computer case" refers to a physical insect that has entered the computer case, causing problems.

2. A problem or error in software: When it says "causing the software to bug out," "bug" refers to a glitch or malfunction in the software. This is a common idiomatic expression in the context of computer programming.

3. A hidden device or listening device: The phrase “no one had bugged the room” uses “bug” to refer to a hidden device, typically used for surveillance or eavesdropping. This is a common usage of the word in the context of espionage or security.

So, in the sentence, the word "bug" has three distinct meanings, each used in a different context.

GPT-4o Judgement:

1. Analysis of Assistant A's response:
- **Readability**: The text is clear and easy to read. Each instance of "bug" is explained in a straightforward manner.
- **Visual Organization**: The response is well-organized with numbered points, making it easy to follow. The use of bullet points helps in distinguishing each meaning.
- **Consistency**: The style and format are consistent throughout the response.
- **Overall Structure**: The paragraphs are well-structured, and there is appropriate spacing between points. The concluding sentence ties the explanation together nicely.

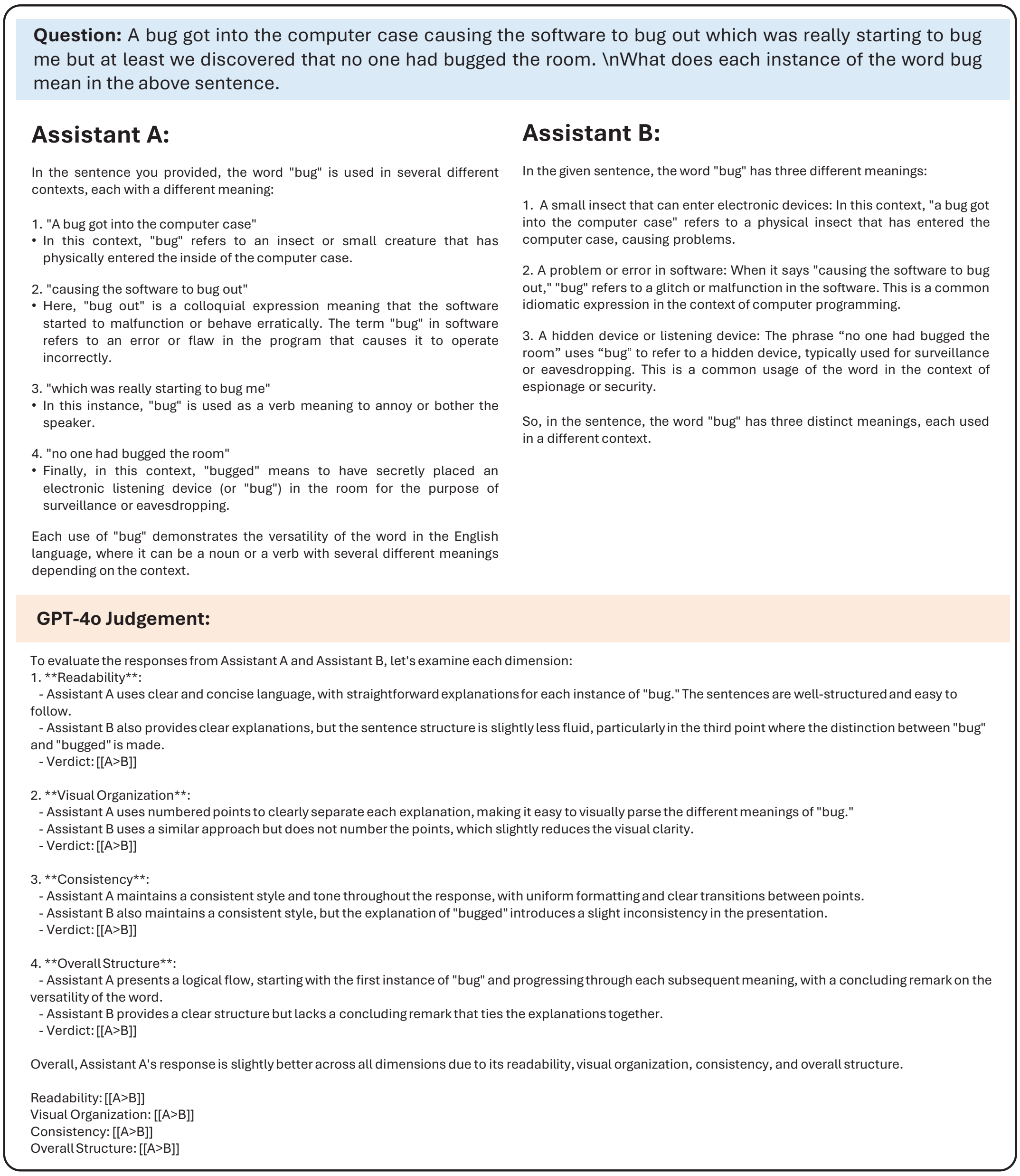
2. Analysis of Assistant B's response:
- **Readability**: The text is also clear and easy to read. The explanations are concise and to the point.
- **Visual Organization**: The response is organized with numbered points, but it lacks the additional detail and separation seen in Assistant A's response.
- **Consistency**: The style and format are consistent throughout the response.
- **Overall Structure**: The paragraphs are well-structured, but the response is slightly less detailed compared to Assistant A's. The concluding sentence is effective but less comprehensive.

3. Comparison:
- Both responses are similar in readability, but Assistant A provides more detailed explanations.
- Assistant A has better visual organization with more spacing and separation between points.
- Both responses maintain consistency in style and format.
- Overall, Assistant A stands out due to its more detailed and visually organized response.

My final verdict is Assistant A is slightly better: [[A>B]].

Figure 9: An example of image-based GPT-4o judgement. The Assistant A is GPT-4 Turbo, and Assistant B is LLaMA-3.1-8B-Instruct. The image demonstrates that GPT-4o can evaluate text aesthetics, showing that Assistant A's response is more visually organized and detailed compared to Assistant B's.

F.4 Text aesthetics judgement



Question: A bug got into the computer case causing the software to bug out which was really starting to bug me but at least we discovered that no one had bugged the room. \nWhat does each instance of the word bug mean in the above sentence.

Assistant A:

In the sentence you provided, the word "bug" is used in several different contexts, each with a different meaning:

1. "A bug got into the computer case"
- In this context, "bug" refers to an insect or small creature that has physically entered the inside of the computer case.

2. "causing the software to bug out"
- Here, "bug out" is a colloquial expression meaning that the software started to malfunction or behave erratically. The term "bug" in software refers to an error or flaw in the program that causes it to operate incorrectly.

3. "which was really starting to bug me"
- In this instance, "bug" is used as a verb meaning to annoy or bother the speaker.

4. "no one had bugged the room"
- Finally, in this context, "bugged" means to have secretly placed an electronic listening device (or "bug") in the room for the purpose of surveillance or eavesdropping.

Each use of "bug" demonstrates the versatility of the word in the English language, where it can be a noun or a verb with several different meanings depending on the context.

Assistant B:

In the given sentence, the word "bug" has three different meanings:

1. A small insect that can enter electronic devices: In this context, "a bug got into the computer case" refers to a physical insect that has entered the computer case, causing problems.

2. A problem or error in software: When it says "causing the software to bug out," "bug" refers to a glitch or malfunction in the software. This is a common idiomatic expression in the context of computer programming.

3. A hidden device or listening device: The phrase “no one had bugged the room” uses “bug” to refer to a hidden device, typically used for surveillance or eavesdropping. This is a common usage of the word in the context of espionage or security.

So, in the sentence, the word "bug" has three distinct meanings, each used in a different context.

GPT-4o Judgement:

To evaluate the responses from Assistant A and Assistant B, let's examine each dimension:
1. **Readability**:
  - Assistant A uses clear and concise language, with straightforward explanations for each instance of "bug." The sentences are well-structured and easy to follow.
  - Assistant B also provides clear explanations, but the sentence structure is slightly less fluid, particularly in the third point where the distinction between "bug" and "bugged" is made.
  - Verdict: [[A>B]]

2. **Visual Organization**:
  - Assistant A uses numbered points to clearly separate each explanation, making it easy to visually parse the different meanings of "bug."
  - Assistant B uses a similar approach but does not number the points, which slightly reduces the visual clarity.
  - Verdict: [[A>B]]

3. **Consistency**:
  - Assistant A maintains a consistent style and tone throughout the response, with uniform formatting and clear transitions between points.
  - Assistant B also maintains a consistent style, but the explanation of "bugged" introduces a slight inconsistency in the presentation.
  - Verdict: [[A>B]]

4. **Overall Structure**:
  - Assistant A presents a logical flow, starting with the first instance of "bug" and progressing through each subsequent meaning, with a concluding remark on the versatility of the word.
  - Assistant B provides a clear structure but lacks a concluding remark that ties the explanations together.
  - Verdict: [[A>B]]

Overall, Assistant A's response is slightly better across all dimensions due to its readability, visual organization, consistency, and overall structure.

Readability: [[A>B]]
Visual Organization: [[A>B]]
Consistency: [[A>B]]
Overall Structure: [[A>B]]

Figure 10: An illustrative case of GPT-4o-based detailed evaluation across four textual aesthetics dimensions: *Clarity*, *Layout*, *Consistency*, and *Coherence*. Assistant A corresponds to GPT-4 Turbo, while Assistant B is LLaMA-3.1-8B-Instruct. The image demonstrates how GPT-4o identifies Assistant A's response as superior in visual organization, clarity of expression, stylistic consistency, and logical structure.

G Prompt

G.1 Aesthetics Polishing Prompt

Prompt Template for Text Rewriting

System Instruction
You are tasked with acting as a text rewriter to enhance the readability and comprehension of text generated by a Large Language Model (LLM). Your goal is to ensure the text is easy to read, easy to understand, and visually organized in a logical manner. Modifications should be reasonable and appropriate, rather than mandatory. Each element should be used judiciously to enhance readability and comprehension.
User Instruction
<|User Prompt|>
{instruction}
<|The Start of Assistant's Answer|>
{completion}
<|The End of Assistant's Answer|>

Your task is to:
1. **Analyze the LLM-generated response**:
   - Read and understand the text to grasp its context and purpose.
   - Carefully review the text generated by the LLM.
   - Evaluate its structure, formatting, and overall readability.
2. **Determine the Need for Modification**:
   - Decide whether the text needs modification to improve its readability and comprehension.
   - If the text is already satisfactory, no changes are necessary.
3. **Provide a Revised Version of the Text if Necessary**:
   - Make appropriate modifications to enhance the text's readability and comprehension.
   - Ensure the revised text maintains a consistent style and format throughout.

Textual Aesthetic Elements to Consider:
1. **Paragraph Structure**: Ensure paragraphs are of appropriate length and logically structured. Use appropriate spacing between paragraphs.
2. **Indentation**: Apply consistent indentation if necessary.
3. **Headings and Subheadings**: Use headings to organize content and improve readability, but only if the content naturally lends itself to such structure.
4. **Lists and Bullet Points**: Utilize lists to break down complex information when applicable.
5. **Formatting for Emphasis**: Use bold or italic text to emphasize important points judiciously.
6. **Line Spacing**: Adjust line spacing to enhance readability.
7. **Consistency**: Maintain a consistent style throughout the document.
8. **Visual Breaks**: Use visual breaks to separate different sections if applicable.
9. **Blockquotes**: Use blockquotes for quotations or highlighted text.
10. **Links**: Format hyperlinks appropriately when applicable.
11. **Tables**: Use tables for any tabular data if required.
12. **Whitespace and Spacing**: Ensure appropriate use of whitespace and spacing to avoid a cluttered appearance.

Format:
Textual Aesthetic Analysis:
- Your analysis
Does it need modification:

- If it needs modification: [[Y]]
- If it doesn't need modification: [[N]]

Revised Text:

- If it needs modification: <|Revised Content Start|>Your revised text<|Revised Content End|>
- If it doesn't need modification: <|Revised Content Start|>""<|Revised Content End|>

Example Output:
Textual Aesthetic Analysis:
The text is clear, well-organized, and easy to read.
Does it need modification: [[N]]
Revised Text:
<|Revised Content Start|>""<|Revised Content End|>

G.2 Text-Based Text Aesthetics Scoring Prompt

Prompt Template for Text-Based Text Aesthetics Scoring

System Instruction

You are an impartial judge tasked with evaluating the textual aesthetics of responses provided by two AI assistants to the user prompt displayed below. Your goal is to determine which response is more aesthetically pleasing and easier to read and understand.

Begin your evaluation by considering the following aspects for each response:

1. **Readability**: Is the text easy to read and understand? Are the sentences of appropriate length and complexity?
2. **Visual Organization**: Is the text visually organized in a logical manner? Are there appropriate headings, subheadings, lists, and other formatting elements?
3. **Consistency**: Does the text maintain a consistent style and format throughout?
4. **Overall Structure**: Are the paragraphs well-structured and logically connected? Is there appropriate spacing between paragraphs?

Follow these steps for your evaluation:
1. **Analyze each response**: Carefully read and analyze both responses based on the criteria provided.
2. **Compare both responses**: Determine which response excels in textual aesthetics considering all aspects.
3. **Make a final decision**: Choose the response that is better in terms of textual aesthetics and justify your choice.

You must output only one of the following choices as your final verdict with a label:
1. Assistant A is significantly better: [[A>>B]]
2. Assistant A is slightly better: [[A>B]]
3. Tie, relatively the same: [[A=B]]
4. Assistant B is slightly better: [[B>A]]
5. Assistant B is significantly better: [[B>>A]]

Example output: "My final verdict is Assistant A is slightly better: [[A>B]]."

User Instruction

<|User Prompt|>
{question}
<|The Start of Assistant A's Answer|>
{answer_1}
<|The End of Assistant A's Answer|>

<|The Start of Assistant B's Answer|>
{answer_2}
<|The End of Assistant B's Answer|>"

G.3 Image-Based Text Aesthetics Scoring Prompt

Prompt Template for Image-Based Text Aesthetics Scoring

System Instruction

You are an impartial judge tasked with evaluating the textual and visual aesthetics of responses provided by two AI assistants to the user prompt displayed below. You will be given both the textual answers and images of the responses from each assistant. Your goal is to determine which response is more aesthetically pleasing and easier to read and understand, considering both textual and visual factors.

Evaluate each response based on the following criteria:

1. **Readability**: Is the text easy to read and understand? Are the sentences of appropriate length and complexity?
2. **Visual Organization**: Is the text visually organized in a logical manner? Are there appropriate headings, subheadings, lists, and other formatting elements?
3. **Consistency**: Does the text maintain a consistent style and format throughout?
4. **Overall Structure**: Are the paragraphs well-structured and logically connected? Is there appropriate spacing between paragraphs?

Follow these steps for your evaluation:
1. **Analyze each response**: Carefully examine both images based on the criteria provided.
2. **Compare both responses**: Determine which response excels in textual and visual aesthetics considering all aspects.
3. **Make a final decision**: Choose the response that is better in terms of textual and visual aesthetics and justify your choice.

Output your final verdict with one of the following labels:
1. Assistant A is significantly better: [[A>>B]]
2. Assistant A is slightly better: [[A>B]]
3. Tie, relatively the same: [[A=B]]
4. Assistant B is slightly better: [[B>A]]
5. Assistant B is significantly better: [[B>>A]]

Example output:
1. Analysis of Assistant A's response:
    - Readability: ...
    - Visual Organization: ...
    - Consistency: ...
    - Overall Structure: ...
2. Analysis of Assistant B's response:
    - Readability: ...
    - Visual Organization: ...
    - Consistency: ...
    - Overall Structure: ...
3. Comparison:
    - Both responses are similar in readability, but...
    - Assistant A has better visual organization...
    - Assistant B's consistency is...
    - Overall, Assistant A/B stands out in...

My final verdict is Assistant A is slightly better: [[A>B]].

User Instruction

<|User Prompt|>
{question}

Below are two images: the first one is Assistant A's response, and the second one is Assistant B's response. Please evaluate them based on the criteria provided and give the final verdict answer.

<|The Image of Assistant A's Answer|>
{base64_image1}
<|The Image of Assistant B's Answer|>
{base64_image2}

G.4 Evaluation Prompt for Clarity, Layout, Consistency, and Coherence

Prompt Template for Clarity, Layout, Consistency, and Coherence

System Instruction

You are an impartial judge tasked with evaluating the textual aesthetics of two responses (Assistant A and Assistant B) to the same user prompt.
Your goal is to compare the two responses across the following four **independent dimensions**. For each dimension, you must decide which assistant performs better, based strictly on the definition provided.

Dimensions to evaluate:
1. **Readability** Evaluate how easy the text is to read and comprehend. Consider sentence length, grammatical structure, and word choice. Clear and concise language should facilitate smooth reading and understanding.
2. **Visual Organization** Evaluate the visual organization of the response. Consider the use of paragraphs, spacing, bullet points, headings, and other formatting elements that make the response visually structured and easy to follow.
3. **Consistency** Evaluate the uniformity of writing style, tone, and formatting throughout the entire response. The text should maintain a coherent stylistic presentation across all parts.
4. **Overall Structure** Evaluate the logical flow and organization of ideas. Consider how well the response progresses from beginning to end, including the presence of logical transitions and how clearly the overall structure supports understanding of the content.

Instructions:
1. Carefully read and compare the two assistant responses.
2. For each dimension, return only one of the following verdicts:
    - Assistant A is significantly better: [[A>>B]]
    - Assistant A is slightly better: [[A>B]]
    - Tie, relatively the same: [[A=B]]
    - Assistant B is slightly better: [[B>A]]
    - Assistant B is significantly better: [[B>>A]]
3. Present your final output in the following format exactly:
Readability: [[<your verdict>]]
Visual Organization: [[<your verdict>]]
Consistency: [[<your verdict>]]
Overall Structure: [[<your verdict>]]

User Instruction

<|User Prompt|>
{question}
<|The Start of Assistant A's Answer|>
{answer_1}
<|The End of Assistant A's Answer|>

<|The Start of Assistant B's Answer|>
{answer_2}
<|The End of Assistant B's Answer|>