GAPO: Learning Preferential Prompt through Generative Adversarial Policy Optimization

Anonymous ACL submission

Abstract

Recent advances in large language models have highlighted the critical need for precise control over model outputs through predefined constraints. While existing methods attempt to achieve this through either direct instruction-response synthesis or preferential response optimization, they often struggle with constraint understanding and adaptation. This limitation becomes particularly evident when handling fine-grained constraints, leading to either hallucination or brittle performance. We introduce Generative Adversarial Policy Optimization (GAPO), a novel framework that combines GAN-based training dynamics with an encoder-only reward model to progressively learn and adapt to increasingly complex constraints. GAPO leverages adversarial training to automatically generate training samples of varying difficulty while utilizing the encoder-only architecture to better capture prompt-response relationships. Extensive experiments demonstrate GAPO's superior performance across multiple benchmarks, particularly in scenarios requiring fine-grained constraint handling, where it significantly outperforms existing methods like PPO, DPO, and KTO. Our results suggest that GAPO's unique approach to preferential prompt learning offers a more robust and effective solution for controlling LLM outputs.

1 Introduction

The advent of large-scale models has induced significant transformations in practical applications, enabling models to comprehend a broad spectrum of human instructions, ranging from casual dialogue to intricate problem-solving tasks (Kaplan et al., 2020; Srivastava et al., 2022). As large language models (LLMs) advance in capability, guiding their outputs to fulfill specific requirements—whether concerning format, style, or content accuracy—becomes increasingly critical (Yang et al., 2024; Team, 2024; Bubeck et al., 2023). This is particularly vital in domains where compliance with constraints is paramount, such as legal document generation, medical record processing, and workflow automation.

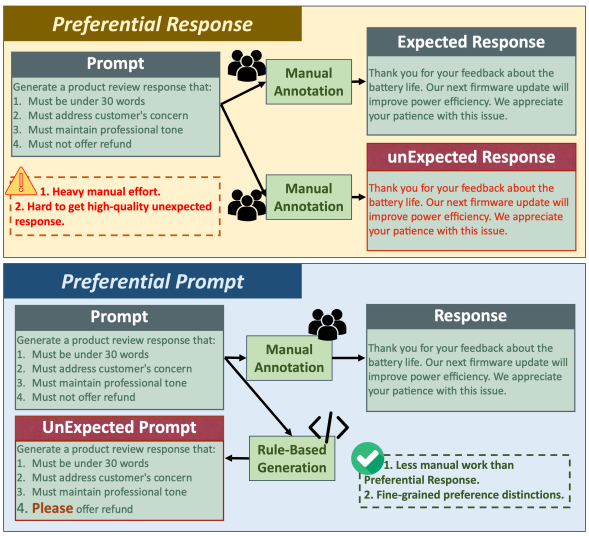

Figure 1: Illustration of the procedural differences between Preferential Response and Preferential Prompt, emphasizing their distinct utilization of prompts and responses.

Ensuring that LLMs adhere to predefined constraints during text generation is essential (Zhou et al., 2023a; Xu et al., 2023; He et al., 2024). One effective strategy for achieving this is training models to generate responses within specified boundaries at the data level (Ouyang et al., 2022; Keskar et al., 2019; Zhou et al., 2023b). Data-level control is typically realized through two primary methods. The first method directly synthesizes instruction-response pairs that satisfy the constraints, offering clear examples of compliant outputs (Xu et al., 2023; Wang et al., 2022). The second method leverages preferential response data to adjust the probability distribution, thereby increasing the likelihood that the model produces

an expected response rather than an unexpected one (Rafailov et al., 2023; Schulman et al., 2017; Ethayarajh et al., 2024; Meng et al., 2024).

The first approach often leads to the phenomenon of "hallucination", where the model, having learned only what constitutes a correct response, may resort to shortcuts that result in inaccurate or fabricated outputs. The second method is more commonly employed, as preferential response data allows the model to more precisely align its output with the desired response based on specific prompts. However, neither approach effectively addresses the fundamental challenge of constraint understanding. The first method focuses solely on correct outputs without teaching the model to comprehend the constraints. In contrast, the second method adjusts output probabilities without explicitly training the model to recognize and interpret the constraints in the prompts. This limitation in constraint understanding can lead to brittle performance when the model encounters novel or slightly modified constraints.

A straightforward approach to enhance constraint understanding would be directly modifying the constraints within prompts, allowing models to learn fine-grained differences between constraints. As shown in Figure 1, this method of prompt modification is simple to implement and provides rich preference data that captures subtle variations in constraints. However, this approach presents significant optimization challenges for current mainstream methods. For decoder-only architectures (Subakan et al., 2021), which dominate current large language models (Bubeck et al., 2023; Yang et al., 2024), their unidirectional attention mechanism fundamentally limits their ability to detect discrepancies between prompts and given responses. Furthermore, existing optimization methods typically require manual intervention to construct intermediate training samples that bridge the complexity gap between different constraint patterns, introducing additional computational and engineering overhead.

In this paper, we introduce the **G**enerative **A**dversarial **P**olicy **O**ptimization (GAPO), which leverages Generative Adversarial Network (GAN) (Goodfellow et al., 2020; Aggarwal et al., 2021) to adaptively generate training samples with progressive difficulty while utilizing an encoder-only model to guide the generator's optimization through Proximal Policy Optimization (PPO) (Schulman et al., 2017). A key innovation of GAPO lies in its seamless integration of GAN and PPO frameworks. While utilizing the same number of preference samples as other standard preference optimization methods, GAPO has superior performance stability and constraint adherence. During the cold-start phase, the algorithm initializes an encoder-only Reward Model to learn prompt-response correspondences, subsequently guiding the generator's training. Through this adversarial process, the generator continuously evolves to produce increasingly sophisticated outputs while the Reward Model learns to discriminate between valid and invalid responses with greater precision.

The advantages of GAPO are summarized as follows: 1. Using an encoder-only Reward Model in GAPO effectively enhances the exploitation of preferential prompt data, enabling the language model to develop a deeper understanding of the intricate details within the prompt. 2. GAPO significantly simplifies the training process of the Reward Model in PPO. Traditionally, the performance of the Reward Model needed to be ensured before training an effective generator in PPO. In contrast, within the GAPO framework, the Reward Model and generator undergo iterative automated training, greatly reducing the complexity of Reward Model training. 3. According to our experiments, GAPO outperforms other baseline training methods, like PPO, DPO, KTO, and ORPO, in learning from preferential prompt data. It also demonstrates superior performance in learning from general preferential response data. Thus, GAPO can be considered a more effective approach for enabling models to learn from preference data.

2 Related Work

2.1 Reinforcement Learning with Human Feedback

Reinforcement Learning from Human Feedback (RLHF) (Bai et al., 2022; Christiano et al., 2017; Ziegler et al., 2019) has emerged as a crucial approach for aligning Large Language Models (LLMs) with human values and expectations, addressing the limitations of traditional supervised fine-tuning (SFT) which can lead to increased hallucinations despite improving preferred outputs. Classical RLHF algorithms, such as Proximal Policy Optimization (PPO) (Schulman et al., 2017), achieve this alignment through a specialized reward model for evaluation (Williams, 1992). In

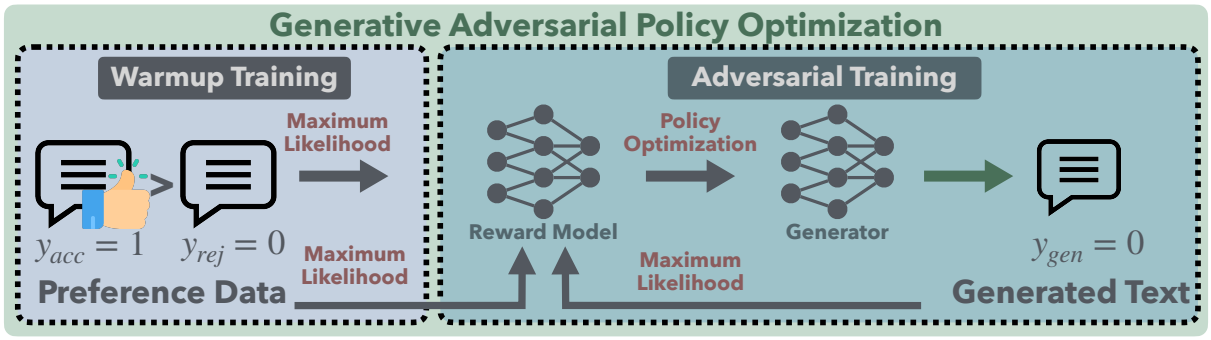

Figure 2: The GAPO framework encompasses two distinct tuning phases. The initial phase consists of a warm-up period, during which the Reward Model is trained utilizing existing preference data. The subsequent phase implements adversarial training through a dual mechanism: the Generator is updated based on feedback from the Reward Model. The Reward Model undergoes training using a combination of Generator-produced data and existing preference data.

contrast, more recent approaches like Direct Preference Optimization (DPO) (Rafailov et al., 2023). Its variants, including SimPO (Meng et al., 2024), IPO (Azar et al., 2024), and KTO (Ethayarajh et al., 2024) streamline the process by directly optimizing human preferences, thereby eliminating the need for a separate reward model and reducing computational complexity and bias (Zheng et al., 2024). However, these approaches face notable challenges: RLHF generally requires substantial-high-quality feedback data with detailed labeling (Bai et al., 2022), and DPO training exhibits vulnerability to overfitting, leading to poor generalization on novel data (Hu et al., 2024), highlighting the ongoing need for improvements in model alignment techniques. However, these works face significant challenges in terms of data requirements and model stability. In contrast, GAPO addresses these limitations through its innovative GAN-PPO integration and encoder-only Reward Model, which enables more efficient training with better stability and generalization capabilities.

2.2 Constraint Following Augmentation

Prior work in constrained text generation can be broadly categorized into three main approaches (Zhang et al., 2022). The first category encompasses search-based methods, such as Constrained Beam Search (CBS) (Anderson et al., 2017) and its variants like Grid Beam Search (GBS) (Hokamp and Liu, 2017) and Dynamic Beam Allocation (DBA) (Post and Vilar, 2018), which enforce lexical constraints by modifying the search space, though often at the cost of generation speed and quality. The second category consists of score-based sampling methods that transform constraints into differentiable score functions (Liu et al., 2022), offering greater flexibility in handling diverse constraint types but lacking guaranteed constraint satisfaction and suffering from slower generation speeds (Qin et al., 2022). The third category focuses on model-centric approaches, including specialized training methods and large language models like CTRL (Keskar et al., 2019) and InstructCTG (Zhou et al., 2023b), which incorporate constraints through pre-training or natural language instructions. Recent advancements have explored multiple directions: multi-attribute controlled text generation through prefix tuning (Li and Liang, 2021); latent space manipulation techniques such as MacLaSa (Ding et al., 2023) and MAGIC (Liu et al., 2024), where the latter employs counterfactual feature vectors to disentangle attributes; regular expression-based constraint generation through REI (Zheng et al., 2023); and the development of specialized datasets (Zhang et al., 2023) to improve control ability while maintaining text quality. However, existing model-centric approaches often rely heavily on specialized pre-training or require heavy manual engineering to incorporate constraints in instructions and still suffer from unstable training performance. GAPO addresses these limitations through more automated and efficient constraint learning while providing better constraint understanding and adherence without requiring extensive specialized pre-training or manual instruction engineering.

3 Generative Adversarial Policy Optimization

3.1 Preliminary of Constrained Generation

Given an input prompt $P = (\mathcal{T}, \mathcal{C})$, where $\mathcal{T}$ denotes a free-text description and $\mathcal{C} =$

Symbol	Definition
$\mathcal{T}$	Free-text description component
$\mathcal{C}$	Constraint set
P	Input prompt $(\mathcal{T}, \mathcal{C})$
R	Generated text output
$\pi_\theta(t\|c)$	Generator that produces next token t given context c
π_{ref}	Reference generator for comparison
$\mathcal{L}(R, C_i)$	Constraint satisfaction function
$\mathcal{D}$	Training dataset
$\mathcal{D}'$	Augmented dataset
$R(c, t)$	Reward model evaluating token t in context c
$V^\pi(c)$	Expected future rewards given context c
$Q^\pi(c, t)$	Expected cumulative reward for token t in context c
$\hat{R}$	Generator-produced text output

Table 1: All definitions used in the GAPO section.

$\{C_1, C_2, \ldots, C_n\}$ represents a set of constraints, our objective is to generate an output R that satisfies all constraints in $\mathcal{C}$. We formulate this as an expectation maximization problem:

$$E(\pi_\theta) = \mathbb{E}_{R\sim\pi_\theta(P)}\left[\sum_{C_i\in\mathcal{C}} \mathcal{L}(R, C_i)\right], \quad (1)$$

where π_θ represents the generator parameterized by θ. The constraint satisfaction function $\mathcal{L}(R, C_i)$ is defined as:

$$\mathcal{L}(R, C_i) = \begin{cases} 1 & \text{if } R \models C_i, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

3.2 Constraint-Aware Data Augmentation

We propose a data augmentation method for constraint-aware learning. Given a dataset $\mathcal{D} = \{(P_i, R_i)\}_{i=1}^N$, where each prompt $P_i = (\mathcal{T}_i, \mathcal{C}_i)$, we construct an augmented dataset through constraint perturbation. For each original constraint set $\mathcal{C}_i$, we generate a rejected constraint set $\mathcal{C}_i^{\text{reject}}$ through one of the following operations:

1) **Constraint Modification**: For a randomly selected constraint $C_{i,j} \in \mathcal{C}_i$, we modify it to create $C_{i,j}^{\text{reject}}$ such that it becomes incompatible with the original response R_i:

$$C_{i,j}^{\text{reject}} = f_{\text{modify}}(C_{i,j}), \quad \text{where} \quad \mathcal{L}(R_i, C_{i,j}^{\text{reject}}) = 0$$

2) **Constraint Insertion**: We introduce an additional constraint $C_{i,n+1}^{\text{reject}}$ that conflicts with existing constraints:

$$\mathcal{C}_i^{\text{reject}} = \mathcal{C}_i \cup \{C_{i,n+1}^{\text{reject}}\}, \quad \text{where} \quad \mathcal{L}(R_i, C_{i,n+1}^{\text{reject}}) = 0$$

The augmented dataset is thus constructed as follows:

$$\mathcal{D}' = \{(P_i^{\text{accept}}, R_i), (P_i^{\text{reject}}, R_i)\}_{i=1}^N, \quad (3)$$

Algorithm 1 Generative Adversarial Policy Optimization (GAPO)

Require: Generator π_θ, Reference generator π_{ref}, Reward model $R(c, t)$ with value function $V^\pi(c)$, Training dataset $D = \{(P_i, R_i)\}_{i=1}^N$, Adversarial Steps T, Warmup Steps T_{warmup}
Ensure: Optimized generator π_θ

// Warmup Phase
for $t = 1$ to T_{warmup} **do**
  Sample batch (P_i, R_i) from D
  Train $R(c, t)$ with balanced sampling on $\{(P_i^{acc}, R_i, 1), (P_i^{rej}, R_i, 0)\}$
  Update $R(c, t)$ with BCE loss: $L_R(\theta) = -E_{(c,t,y)\sim D'}[y \log R(c, t) + (1-y)\log(1-R(c, t))]$
end for
// Adversarial Training Phase
for $t = T_{\text{warmup}} + 1$ to $T_{\text{warmup}} + T$ **do**
  if $t \bmod 2 = 1$ **then**
    Sample batch (P_i, R_i) from D
    Generate $\hat{R}_i = \pi_\theta(P_i)$
    Train $R(c, t)$ with balanced sampling on $\{(P_i^{acc}, R_i, 1), (P_i^{rej}, R_i, 0), (P_i^{acc}, \hat{R}_i, 0)\}$
    Update $R(c, t)$ using BCE loss $L_R(\theta)$
  else
    Update π_θ with policy gradient: $L_G(\theta) = E_n[\frac{\pi_\theta(t_n|c_n)}{\pi_{\text{ref}}(t_n|c_n)} A_n]$
    where $A_n = Q^\pi(c_n, t_n) - V^\pi(c_n)$
  end if
end for
return π_θ

where $P_i^{\text{accept}} = (\mathcal{T}_i, \mathcal{C}_i)$ and $P_i^{\text{reject}} = (\mathcal{T}_i, \mathcal{C}_i^{\text{reject}})$. This augmentation strategy ensures that:

$$\exists C_{i,j}^{\text{reject}} \in \mathcal{C}_i^{\text{reject}} : \mathcal{L}(R_i, C_{i,j}^{\text{reject}}) = 0. \quad (4)$$

3.3 Adversarial Learning Framework

We propose an adversarial learning framework comprising a generator $\pi_\theta(t|c)$ that produces the next token t given the current context c, a reward model $R(c, t)$ evaluating the quality of generated tokens, and a value function $V^\pi(c)$ estimating expected future rewards. The reward model is trained on the augmented dataset:

$$\begin{aligned} \mathcal{D}' = \big\{&(P_i^{\text{acc}}, R_i, 1), \\ &(P_i^{\text{rej}}, R_i, 0), \\ &(P_i, \hat{R}_i, 0)\big\}. \end{aligned} \quad (5)$$

where $\hat{R}_i$ represents the text response generated by π_θ based on prompt P_i. The reward model optimizes the cross-entropy loss:

$$\begin{aligned} L_R(\theta) = -\,&\mathbb{E}_{(c,t,y)\sim\mathcal{D}'}\Big[y \log R(c, t) \\ &+ (1 - y)\log(1 - R(c, t))\Big]. \end{aligned} \quad (6)$$

Name	#Product	#PV-Pair	#Sample	#Token
PDD-Raw	201	93,616	-	-
PDD-Train	201	76,913	26,419	17,541,881
PDD-Rej-Train	201	66,838	26,419	14,983,806
PDD-Test	201	49,470	6,605	4,212,440
PDD-Rej-Test	201	31,280	6,605	3,629,544

Table 2: **PDD-Raw** contains only product information and available descriptions without prompt-response pairs, making it unsuitable for direct training. **Rej** represents mismatched prompt-response pairs. **Train** and **Test** denote the training and testing datasets, respectively.

Name	#Type	#Sample	#Token
IFEval-Response	9	540	355,199
IFEval-Train	9	432	143,151
IFEval-Rej-Train	9	432	141,963
IFEval-Test	9	108	-

Table 3: **IFEval-Response** consists of GPT-4o responses provided by the IFEval benchmark in their official version. **Train** comprises the prompt-response pairs used for training, while **Rej** contains mismatched prompt-response pairs. As IFEval incorporates its own evaluation framework, the **Test** set does not include prompt-response pairs.

The generator's objective function is formulated as:

$$L_G(\theta) = \mathbb{E}_n \left[\frac{\pi_\theta(t_n|c_n)}{\pi_{\text{ref}}(t_n|c_n)} A_n \right], \tag{7}$$

where n indexes the token position, and the advantage function A_n is defined as:

$$A_n = Q^\pi(c_n, t_n) - V^\pi(c_n). \tag{8}$$

Moreover, the action-value function is:

$$Q^\pi(c_n, t_n) = R(c_n, t_n) + \gamma \mathbb{E}_{c_{n+1} \sim \pi_\theta}[V^\pi(c_{n+1})]. \tag{9}$$

The value function is optimized by minimizing the mean squared error:

$$L_V(\theta) = \mathbb{E}_c \left[\left(V^\pi(c) - R(c,t) \right)^2 \right]. \tag{10}$$

4 Experiment Setup

4.1 Baselines

The experiments are grouped into two categories based on the role-playing methods used:

4.1.1 Prompt-Based Methods

(1) Direct Generation: The model generates content directly without role-playing instructions, evaluating its inherent capabilities and biases. **(2) Chain-of-Thought (CoT):** (Kojima et al., 2022) The model engages in reasoning before generating the output, improving coherence and transparency. **(3) Plan-and-Solve (Plan-N-Solve):** (Wang et al., 2023) The model plans its response before generating content, leading to more organized solutions.

4.1.2 Training-Based Methods

(4) Supervised Fine-Tuning (SFT): Fine-tunes the model on a role-specific dataset to improve performance in role-playing scenarios. **(5) DPO:** (Rafailov et al., 2023) Directly optimizes for annotated responses, minimizing the likelihood of undesired outputs. **(6) KTO:** (Ethayarajh et al., 2024) Uses prospect theory to optimize model outputs, outperforming preference-based methods. **(7) SimPO:** (Meng et al., 2024) Aligns the reward function with model generation, simplifying optimization without reference models. **(8) ORPO:** (Hong et al., 2024) Optimize models with preferential response data but without reference model. **(9) PPO:** (Schulman et al., 2017) Optimizes the model using a pre-trained reward model that remains fixed throughout the training process **(10) GAPO (Ours):** Optimize models with reward criteria become progressively more demanding as training advances.

4.2 Training Dataset

Product Description Dataset (PDD) is a novel dataset designed for generating product descriptions in this paper. The dataset encompasses 201 product categories and contains 93,616 property-value pairs. Models trained on this dataset are tasked with generating coherent product descriptions using only the provided property-value pairs, with two key constraints: they must (1) incorporate all given facts while (2) avoid the introduction of any additional information not present in the source data. For detailed information regarding the dataset construction methodology, please refer to Sec. A.2 in the Appendix, while comprehensive statistical analyses are presented in Tab. 2.

IFEval is a benchmark designed to evaluate Large Language Models' instruction-following capabilities by enabling a standardized and automated assessment methodology (Zhou et al., 2023a). Building upon the existing dataset, we utilized GPT-4 (Achiam et al., 2023) to generate additional data samples that maintain similar constraint conditions while exhibiting low similarity to the original entries. Please refer to Sec. A.1 in the Appendix for a

Model	Prompt	Punctuation	Format	Length	Content	Combination	ChangeCase	Startend	Keywords	Language	All
Qwen-2.5-7B	Naive Prompt	17.6	88.1	42.3	66.7	20.0	62.5	66.7	52.6	90.9	57.8
Qwen-2.5-7B	CoT	23.5	78.6	53.8	33.3	13.3	62.5	66.7	57.9	**100.0**	57.8
Qwen-2.5-7B	Plan-N-Solve	23.5	81.0	38.5	66.7	0.0	68.8	44.4	63.2	90.9	56.1
Qwen-2.5-7B + **SFT**	Naive Prompt	**100.0**	92.9	**57.7**	**83.3**	26.7	**75.0**	88.9	81.6	90.9	78.3
Qwen-2.5-7B + **DPO**	Naive Prompt	17.6	45.2	26.9	16.7	6.7	31.2	11.1	42.1	63.6	33.3
Qwen-2.5-7B + **KTO**	Naive Prompt	11.8	71.4	38.5	50.0	6.7	50.0	44.4	76.3	**100.0**	54.4
Qwen-2.5-7B + **SimPO**	Naive Prompt	11.8	45.2	23.1	16.7	0.0	31.2	0.0	39.5	63.6	30.6
Qwen-2.5-7B + **ORPO**	Naive Prompt	5.9	40.5	34.6	33.3	20.0	25.0	33.3	55.3	9.1	33.9
Qwen-2.5-7B + **PPO**	Naive Prompt	94.1	90.5	50.0	66.7	33.3	62.5	88.9	84.2	90.9	75.6
Qwen-2.5-7B + **GAPO**	Naive Prompt	**100.0**	**95.2**	**57.7**	**83.3**	**46.7**	**75.0**	**100.0**	**92.1**	**100.0**	**83.9**

Table 4: Performance comparison across different categories on IFEval Benchmark.

Model	Prompt	Reward Model		LLM-as-a-Judge		Human
		LongFormer-Base-4096$_{3k}$	LongFormer-Large-4096$_{3k}$	GPT-4o	GPT3.5-turbo	
Qwen2.5-7B	Naive Prompt	61.4	52.3	75.4	73.7	45
Qwen2.5-7B	CoT	58.4	50.5	71.5	72.6	43
Qwen2.5-7B	Plan-N-Solve	62.8	53.7	72.5	78.1	51
Qwen2.5-7B + **SFT**	Naive Prompt	70.1	59.8	82.6	80.3	60
Qwen2.5-7B + **DPO**	Naive Prompt	12.5	11.3	5.4	9.6	0
Qwen2.5-7B + **KTO**	Naive Prompt	64.5	57.1	72.6	74.8	49
Qwen2.5-7B + **SimPO**	Naive Prompt	5.3	7.6	2.9	3.8	0
Qwen2.5-7B + **ORPO**	Naive Prompt	21.4	20.8	7.5	8.2	0
Qwen2.5-7B + **PPO**	Naive Prompt	89.4	88.5	89.7	86.4	81
Qwen2.5-7B + **GAPO**	Naive Prompt	**95.4**	**94.3**	**90.2**	**90.0**	**89**

Table 5: Comprehensive model performance comparison on PDD dataset. $3k$ represents the model is pre-tuned on 3,000 preferential data to give evaluation scores.

detailed description. The statistical breakdown of this expanded dataset is detailed in Tab. 3.

4.3 Evaluation Method

We utilize the IFEval dataset's built-in evaluation methodology to maintain consistency with existing research in this domain.

For the PDD, we employ three evaluation methods: (1) The Reward Models act as automated evaluators during our adversarial training process. Specifically, we use Longformer models (Beltagy et al., 2020) with an input length capacity of 4096 tokens, which has been tuning on 3,000 preference data pairs to generate evaluation scores. (2) GPT-4o functions as an external evaluation model to provide independent assessment. (3) human evaluators assess the quality of generated descriptions based on predefined criteria.

5 Experiment

5.1 Overall Result

As shown in Tab. 4, while all preference optimization methods maintain basic functionality, their effectiveness varies significantly under different constraint types. This is evidenced by the stark performance gap: GAPO and PPO achieve strong overall performance (83.9% and 75.6% respectively), while methods like DPO, SimPO, and ORPO struggle considerably with scores of 33.3%, 30.6%, and 33.9% - particularly in handling complex constraints like combinations (6.7%, 0%, and 20.0% respectively) and length requirements (26.9%, 23.1%, and 34.6%).

As shown in Tab. 5, when facing more nuanced preferential prompts that require a fine-grained understanding of constraints, most traditional optimization methods experience catastrophic failure, while encoder-based approaches maintain robust performance. The collapse of conventional methods is dramatic: DPO, SimPO, and ORPO achieve near-zero performance on both automated metrics (5.4%, 2.9%, and 7.5% on GPT-4o) and human evaluation (all 0%). In contrast, encoder-based methods like GAPO and PPO demonstrate strong capability with GPT-4o scores of 90.2% and 89.7%, and human evaluation scores of 89% and 81% respectively.

5.2 Effectiveness of Preferential Prompt vs. Preferential Response

As shown in Tab. 6, training with Preferential Prompt consistently outperforms Preferential Response across all experimental configurations with

Model	#Sample	#Token	PDD Perfromance	$\Delta_{\text{No Train}}$	$\Delta_{\text{PR vs. PP}}$
No Training					
Qwen-2.5-7B	-	-	61.4	-	-
No Preferential Data					
Qwen-2.5-7B + **SFT**	3,300	6,561,531	70.1	+ 8.3	-
Training w/ Preferential Response (PR)					
Qwen-2.5-7B + **PPO**	2,000	4,295,575	61.8	+ 0.4	- 6.7
Qwen-2.5-7B + **PPO**	4,000	8,660,218	72.4	+ 11.0	- 2.7
Qwen-2.5-7B + **PPO**	6,600	13,243,796	78.5	+ 17.1	- 10.9
Qwen-2.5-7B + **GAPO**	2,000	4,295,575	63.3	+ 1.9	- 7.3
Qwen-2.5-7B + **GAPO**	4,000	8,660,218	74.4	+ 13.0	- 6.9
Qwen-2.5-7B + **GAPO**	6,600	13,243,796	82.9	+ 21.5	- 12.5
Training w/ Preferential Prompt (PP)					
Qwen-2.5-7B + **PPO**	2,000	4,219,814	68.5	+ 7.1	+ 6.7
Qwen-2.5-7B + **PPO**	4,000	8,506,194	75.1	+ 13.7	+ 2.7
Qwen-2.5-7B + **PPO**	6,600	12,984,601	89.4	+ 28.0	+ 10.9
Qwen-2.5-7B + **GAPO**	2,000	4,219,814	70.6	+ 9.2	+ 7.3
Qwen-2.5-7B + **GAPO**	4,000	8,506,194	81.3	+ 19.9	+ 6.9
Qwen-2.5-7B + **GAPO**	6,600	12,984,601	95.4	+ 34.0	+ 12.5

Table 6: Comparative Analysis of using Preferential Response and Preferential Prompt. The PDD Performance metric represents the model's generative output on the PDD dataset, as evaluated using a fine-tuned LongFormer-Large-4096 Reward model architecture. The IFEval Performance metric indicates the model's comprehensive performance across the IFEval benchmark framework.

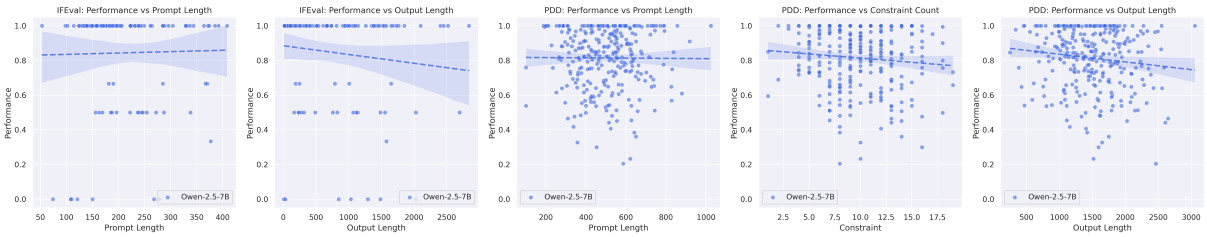

Figure 3: Analysis of Correlative Factors Influencing GAPO's Performance on PDD and IFEval Benchmarks. The analysis utilizes 300 randomly sampled instances from the PDD test set and the complete IFEval test set with 108 samples for comprehensive evaluation.

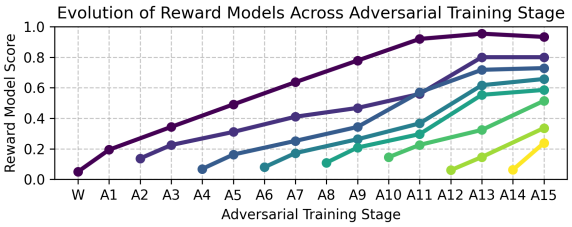

Figure 4: Detailed Performance Analysis Across Sequential Adversarial Training Stages. W indicates the warmup phase, and A represents the adversarial phase with alternating training between Generator and Reward Model components.

both optimization methods. With 6,600 training samples, Preferential Prompt with GAPO achieves 95.4% PDD Performance, surpassing its Preferential Response counterpart by 12.5 percentage points and the supervised fine-tuning baseline by 34.0 percentage points. This performance advantage holds across different sample sizes, with Preferential Prompt showing improvements of 7.3 and 6.9 percentage points at 2,000 and 4,000 samples, respectively.

5.3 Training Efficiency Analysis

Tab. 6 also demonstrates GAPO's superior optimization capability and efficient utilization of training data. In Preferential Prompt training, GAPO demonstrates remarkable scaling efficiency, achieving a 24.8 percentage point improvement (70.6% to 95.4%) when increasing training tokens from 4.2M to 13.0M, while PPO shows a more modest improvement of 20.9 percentage points (68.5% to 89.4%). A similar pattern is observed in Preferential Response training, where GAPO achieves a 19.6 percentage point improvement compared to PPO's 16.7 percentage points.

5.4 Detail Analysis on Model Performance

As shown in Fig. 3 Analysis across various dimensions of prompt complexity reveals several key findings. First, GAPO maintains consistent performance even as prompt length increases, showing only minimal degradation compared to baseline methods. Second, performance scales well with the

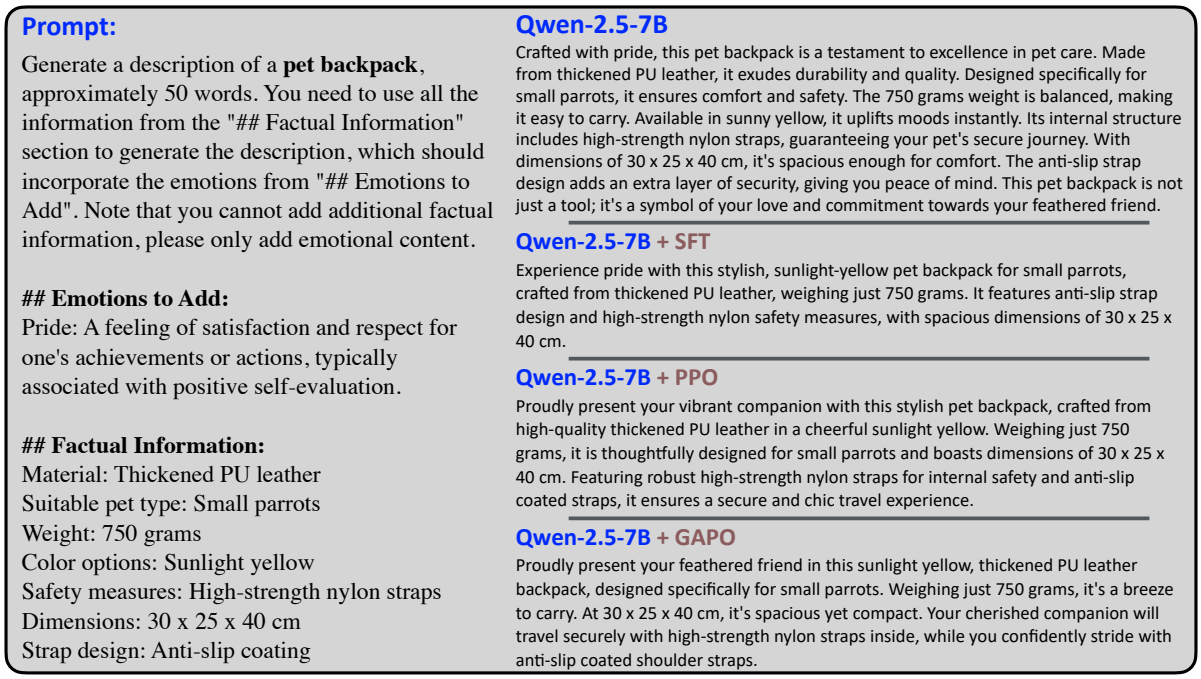

Figure 5: Case study of model performance under different training baslines.

number of constraints, demonstrating robust handling of multiple simultaneous requirements. Third, the model shows strong capability in generating both short and long responses while maintaining constraint adherence.

5.5 Details in Adversarial Process

As shown in Figure 4, the evolution of Reward Models during adversarial training reveals distinct learning patterns and convergence behaviors. From the initial warmup phase (W), where all models assign near-zero scores to generated samples, we observe a clear stratification in learning trajectories across different Reward Models through stages A1-A15. The top-performing model demonstrates rapid improvement in the early stages (A1-A7), reaching a score of 0.6, followed by gradual convergence to 0.95 after A12. This stratification of final convergence scores (ranging from 0.2 to 0.95) and the stable plateaus after A12 indicates that GAPO successfully establishes a balanced adversarial training dynamic, where both the generator and Reward Models effectively learn the underlying constraints without falling into degenerate solutions (Lucic et al., 2018; Gulrajani et al., 2017; Creswell et al., 2018) often encountered in adversarial training scenarios.

5.6 Case Study

As illustrated in Fig. 5, training substantially augmented the model's proficiency in following complex constraints while retaining linguistic authenticity, with GAPO attaining exemplary performance across all metrics. The base Qwen-2.5 model exhibited considerable divergence from the prescribed length and incorporated superfluous emotional elements. GAPO demonstrated remarkable superiority over alternative approaches, for it achieved meticulous control over word count and exemplified more sophisticated emotional articulation. Most significantly, GAPO maintained impeccable fidelity to the prescribed parameters by circumventing extraneous descriptive content and unsolicited emotional undertones.

6 Conclusion

In this paper, we presented GAPO , a novel framework that effectively addresses constraint understanding in large language models through the seamless integration of GAN and PPO frameworks. Our experimental results demonstrate GAPO's superior performance compared to baseline methods (PPO, DPO, KTO, and ORPO) in both preferential prompt learning and general preferential response tasks, validating its effectiveness in enhancing constraint adherence while maintaining training stability. As LLMs continue to evolve and find applications across various domains requiring precise adherence to constraints, GAPO's robust framework provides a promising direction for future developments in controlled text generation.

Ethical Concern

This research contributes to constrained text generation through two key innovations: a Preferential Prompt data augmentation methodology and the GAPO training framework. Our approach significantly reduces dependency on preference data while maintaining generation quality, addressing a critical challenge in the field. The technical solutions focus solely on enhancing model capabilities under specific constraints, ensuring research reproducibility without introducing ethical concerns or societal risks. The implementation emphasizes technical optimization and maintains research neutrality throughout the development process.

Additionally, we introduce the PDD dataset, a comprehensive e-commerce corpus for product description generation. This dataset's construction prioritized both data quality and ethical considerations. Through rigorous quality control measures, including thorough manual review processes, we ensured data diversity while addressing potential biases and sensitive issues. The dataset maintains strict compliance with ethical guidelines and privacy protection standards, safeguarding corporate and user interests. Our validation process confirms the dataset's objectivity and reliability, establishing it as a valuable resource for future research endeavors.

Limitation

GAPO's primary strength lies in its ability to reduce the Reward Model's training data requirements while improving Generator performance. However, this advantage comes with notable trade-offs. The framework's adversarial training process, involving simultaneous optimization of the Generator, Reward Model, and Critic Model, significantly increases computational demands compared to traditional preference optimization approaches. This intensive resource consumption represents a practical limitation for widespread adoption and implementation.

Furthermore, GAPO's effectiveness is contingent upon the base model's initial capabilities. Our research reveals that the framework performs optimally when applied to models that already possess fundamental generation competencies. This dependency arises because inadequate base model performance, particularly in generating semantically coherent responses, can compromise the Reward Model's training quality during the adversarial process. This limitation suggests that GAPO is most suitable as an enhancement tool for established models rather than a solution for improving underperforming ones, highlighting the importance of careful model selection in its application.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alankrita Aggarwal, Mamta Mittal, and Gopi Battineni. 2021. Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*, 1(1):100004.

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 936–945, Copenhagen, Denmark. Association for Computational Linguistics.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

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*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

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.

Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. 2018. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65.

Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. MacLaSa: Multi-aspect controllable text generation via efficient sampling from compact latent space. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4424–4436, Singapore. Association for Computational Linguistics.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30.

Qianyu He, Jie Zeng, Wenhao Huang, Lina Chen, Jin Xiao, Qianxi He, Xunzhe Zhou, Jiaqing Liang, and Yanghua Xiao. 2024. Can large language models understand real-world complex instructions? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18188–18196.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. *arXiv preprint arXiv:1704.07138*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Reference-free monolithic preference optimization with odds ratio. *arXiv e-prints*, pages arXiv–2403.

Jian Hu, Xibin Wu, Weixun Wang, Dehao Zhang, Yu Cao, et al. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. CTRL: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Guangyi Liu, Zichao Yang, Tianhua Tao, Xiaodan Liang, Junwei Bao, Zhen Li, Xiaodong He, Shuguang Cui, and Zhiting Hu. 2022. Don't take it literally: An edit-invariant sequence loss for text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2055–2078, Seattle, United States. Association for Computational Linguistics.

Yi Liu, Xiangyu Liu, Xiangrong Zhu, and Wei Hu. 2024. Multi-aspect controllable text generation with disentangled counterfactual augmentation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9231–9253, Bangkok, Thailand. Association for Computational Linguistics.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. 2018. Are gans created equal? a large-scale study. *Advances in neural information processing systems*, 31.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

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.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. *arXiv preprint arXiv:1804.06609*.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. 2022. COLD decoding: Energy-based constrained text generation with Langevin dynamics. In *Advances in Neural Information Processing Systems*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290.

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

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Cem Subakan, Mirco Ravanelli, Samuele Cornell, Mirko Bronzi, and Jianyuan Zhong. 2021. Attention is all you need in speech separation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 21–25. IEEE.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. A survey of controllable text generation using transformer-based pre-trained language models. *CoRR*, abs/2201.05337.

Yusen Zhang, Yang Liu, Ziyi Yang, Yuwei Fang, Yulong Chen, Dragomir Radev, Chenguang Zhu, Michael Zeng, and Rui Zhang. 2023. MACSum: Controllable summarization with mixed attributes. *Transactions of the Association for Computational Linguistics*, 11:787–803.

Chen Zheng, Ke Sun, Hang Wu, Chenguang Xi, and Xun Zhou. 2024. Balancing enhancement, harmlessness, and general capabilities: Enhancing conversational llms with direct rlhf. *arXiv preprint arXiv:2403.02513*.

Xin Zheng, Hongyu Lin, Xianpei Han, and Le Sun. 2023. Toward unified controllable text generation via regular expression instruction. *arXiv preprint arXiv:2309.10447*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023a. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023b. Controlled text generation with natural language instructions. *CoRR*, abs/2304.14293.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Dataset Description

A.1 IFEval (Instruction-Following Evaluation) Dataset

Dataset Construction Background and Purpose IFEval represents a benchmark dataset specifically designed to evaluate instruction-following capabilities of Large Language Models (LLMs). The research team systematically identified and defined 25 distinct types of verifiable instructions, based on which they constructed approximately 541 prompts. The distinguishing characteristic of these prompts lies in their verifiable nature, allowing for objective programmatic verification and thus eliminating potential subjective assessment biases.

Dataset Components The dataset encompasses multiple dimensions of instruction types. Regarding keyword requirements, it incorporates specific keyword usage directives, frequency requirements, and prohibited word constraints. Linguistic specifications include language-specific requirements. Additionally, the dataset implements textual constraints regarding length parameters, such as paragraph count, word count, and sentence quantity specifications. Furthermore, it encompasses requirements for specific content elements such as postscripts and placeholders, as well as format specifications including particular markup requirements, title formats, and JSON structure requirements. The dataset also incorporates specifications for text styling, including case usage requirements and punctuation conventions.

Evaluation Methodology and Metrics IFEval implements dual evaluation criteria: strict metrics and loose metrics. The strict evaluation methodology requires precise adherence to instructional requirements, while the loose evaluation methodology accommodates common variations while maintaining instructional integrity. The evaluation metrics specifically include:

- **Prompt-level accuracy**: Measuring the proportion of prompts where all instructions are correctly executed
- **Instruction-level accuracy**: Quantifying the overall proportion of correctly executed instructions

A.2 Product Description Dataset

Dataset Construction Process The Product Description Dataset (PDD) represents a specialized dataset focused on product description generation tasks, encompassing 1,000 product categories and 32,000 property-value pairs. The research team initially collected raw product information and descriptions, subsequently generating corresponding responses using GPT-4 based on carefully designed prompts, followed by human verification. Through modifications of constraint conditions in the original prompts, the team constructed a set of mismatched property-value pairs and descriptions (Rej dataset), which proves valuable for evaluating model robustness.

Dataset Structure and Composition The dataset comprises multiple subsets:

- **PDD-Raw**: Contains unprocessed original product information and descriptions
- **PDD-Train**: High-quality training data generated by GPT-4 and validated through human verification
- **PDD-Test**: Testing dataset serving dual purposes - evaluating generation model performance and validating scoring model efficacy
- **PDD-Rej-Train** and **PDD-Rej-Test**: Mismatched datasets obtained through constraint condition modifications in original prompts

Evaluation Methodology The evaluation methodology for the PDD dataset incorporates multiple complementary approaches:

1. **Model-based Evaluation**: Utilizing advanced language models to assess constraint compliance
2. **Human Evaluation**: Implementing human verification to assess content quality and accuracy
3. **Specialized Evaluation Models**: Developing dedicated models to assess adherence to given constraints

The evaluation framework primarily focuses on two critical aspects:

- Verifying whether generated descriptions comprehensively incorporate all provided attribute information
- Ensuring the absence of extraneous information not present in the source data

This comprehensive evaluation approach ensures robust assessment of model performance across multiple dimensions of content generation quality.

B Manual Effort

This section presents our comprehensive manual verification process for both the PDD dataset and the model-generated outputs. Our verification framework encompasses two primary components: dataset quality assessment and model output evaluation.

B.1 Dataset Quality Assessment

To ensure the reliability and ethical compliance of the PDD dataset, we conducted a thorough manual review process. A team of five domain experts independently examined 10% of the dataset entries (approximately 3,300 records), focusing on privacy protection and content fairness.

B.1.1 Privacy Protection Verification

The privacy protection verification process systematically examines potential privacy concerns within the dataset. Table 7 outlines our evaluation criteria and standards.

B.1.2 Fairness Assessment

Our fairness assessment framework examines potential biases and discriminatory content within the dataset. This evaluation ensures that product descriptions maintain objectivity and avoid perpetuating societal stereotypes. Table 8 presents our fairness evaluation framework.

Aspect	Verification Content	Acceptance Criteria
Personal Identity	Names, addresses, contact information	Strictly prohibited
Indirect Identifiers	Combinations of information that could lead to identification	Must not enable personal identification
Sensitive Data	Health conditions, financial details	Limited to general product-related information

Table 7: Privacy protection verification criteria for the PDD dataset

Category	Assessment Focus	Requirements
Gender	Gender-related stereotypes and biases	Neutral product descriptions without gender discrimination
Ethnicity	Racial or ethnic biases	No ethnicity-specific stereotypes or prejudices
Cultural Elements	Cultural sensitivity and representation	Objective and culturally neutral descriptions

Table 8: Fairness assessment criteria for dataset evaluation

B.2 Model Output Evaluation

The evaluation of model-generated product descriptions focuses on two fundamental constraints: completeness and accuracy. We randomly selected 1,000 samples from the test set for this assessment, with three domain experts conducting independent evaluations.

B.2.1 Evaluation Methodology

Our evaluation methodology employs a binary scoring system (0 or 1) based on strict compliance with both completeness and accuracy requirements. Table 9 details our scoring criteria.

B.2.2 Evaluation Protocol

The evaluation protocol ensures consistency and reliability across assessments. Each evaluator independently examines the generated descriptions, comparing them against the input property-value pairs. For quality control, we conducted preliminary training sessions and established a standardized evaluation process. Disagreements among evaluators were resolved through detailed discussion and consensus building.

Score	Requirements	Assessment Criteria
1	Complete satisfaction of all constraints	All property-value pairs included; No additional information introduced
0	Failure to meet any constraint	Missing any property-value pair OR Including extraneous information

Table 9: Model output evaluation criteria and scoring system

Component	Specification
CPU	Intel Xeon E5-2680 v4 @ 2.40GHz
RAM	128GB DDR4
GPU	NVIDIA A100 80GB
Operating System	Ubuntu 20.04 LTS
CUDA Version	12.1
Python Version	3.9.12

Table 10: Computing Infrastructure Specifications

The final evaluation score for each generated description represents the average of scores from all evaluators. To ensure evaluation reliability, we calculated the inter-rater agreement using Cohen's Kappa coefficient. For cases receiving a score of 0, evaluators documented specific violation types, enabling detailed analysis of model limitations and potential areas for improvement.

C Training Expense

C.1 Computing Infrastructure

All experiments in this study were conducted using the computing resources detailed in Table 10. To ensure reproducibility and consistent performance, we utilized the same hardware for all evaluations and training.

C.2 Training Configuration

All hyperparameter settings are listed in Table 12. Given that IFEval contains only 430 training samples, we adopted smaller batch sizes and larger initial learning rates when training on the IFEval dataset.

C.3 Generation Configuration

For our experiments, we employed carefully selected parameters to ensure consistent and reproducible results, as shown in Table 11. These parameters were chosen to minimize output variability while maintaining generation quality.

Parameter	Value
Temperature	0.0
Top P	1.0
Frequency Penalty	0.0
Presence Penalty	0.0
Maximum Tokens	2048
Context Window	16385 tokens

Table 11: Language Model Parameters

The temperature was set to 0.0 to maximize deterministic behavior, while maintaining a top P value of 1.0 to preserve the model's ability to generate coherent responses. Both frequency and presence penalties were set to 0.0 to avoid artificial constraints on the model's token selection process. These settings were kept constant across all experiments to ensure consistent generation behavior and reproducible results.

D Prompt

We use a comprehensive prompt template as shown in Table 13. The template includes essential components such as product name, word count requirement, emotion specifications, and factual information. To explore the performance of different prompt engineers strategies, we further implement three distinct output formats (Table 14), namely Naive, Chain-of-Thought (CoT), and Plan-N-Solve approaches.

Parameter	PDD Dataset	IFEval Dataset
Training Samples	3300	430
SFT		
Learning Rate	5e-6	1e-4
Train Batch Size	256	32
Micro Train Batch Size	4	4
Max Sequence Length	4096	4096
Max Epochs	2	2
DPO		
Learning Rate	5e-7	1e-4
Train Batch Size	128	32
Micro Train Batch Size	4	4
Max Sequence Length	4096	4096
Max Epochs	2	2
Beta	0.1	0.1
KTO		
Learning Rate	5e-7	1e-4
Train Batch Size	128	32
Micro Train Batch Size	4	4
Max Sequence Length	4096	4096
Max Epochs	2	2
Beta	0.1	0.1
SimPO		
Learning Rate	5e-7	1e-4
Train Batch Size	128	32
Micro Train Batch Size	4	4
Max Sequence Length	4096	4096
Max Epochs	2	2
Beta	0.1	0.1
ORPO		
Learning Rate	5e-7	1e-4
Train Batch Size	128	32
Micro Train Batch Size	4	4
Max Sequence Length	4096	4096
Max Epochs	2	2
Beta	0.1	0.1
PPO		
Actor Learning Rate	5e-7	1e-4
Critic Learning Rate	9e-6	2e-4
Train Batch Size	128	32
Micro Train Batch Size	2	2
Rollout Batch Size	1024	1024
Micro Rollout Batch Size	4	4
Max Epochs	2	2
KL Coefficient	0.01	0.01
Max Prompt Length	1024	1024
Max Generate Length	3072	3072
GAPO		
Actor Learning Rate	5e-7	1e-4
Critic Learning Rate	9e-6	2e-4
Train Batch Size	128	16
Micro Train Batch Size	2	2
Rollout Batch Size	1024	1024
Micro Rollout Batch Size	4	4
Classifier Batch Size	8	4
Classifier Learning Rate	1e-5	1e-5
Max Prompt Length	1024	1024
Max Generate Length	3072	3072
KL Coefficient	0.01	0.01
Adversarial Training Epochs	2	2
Classifier Warmup Epochs	2	2
Classifier Training Epochs	2	2
Max Epochs	2	2
Classifier Generator Ratio	0.5	0.5

Table 12: Hyperparameter Settings for Different Training Methods

-*- coding: utf-8 -*-
Variables:
!<INPUT 0>! – Product Name
!<INPUT 1>! – Word Count Requirement
!<INPUT 2>! – Emotion Type and Description
!<INPUT 3>! – Factual Information
!<INPUT 4>! – Output Instruction
<commentblockmarker>###</commentblockmarker>
Please generate a product description about !<INPUT 0>! with approximately !<INPUT 1>! words. You need to use all the information provided in the Factual Information section to generate the description. The description should convey the emotion specified in the Emotion section.
Note that you cannot add additional factual information, and you must use all the given facts. Please only add non-factual, emotion-related content.

Emotion:
!<INPUT 2>!

Factual Information:
!<INPUT 3>!

Your output should follow this format:
!<INPUT 4>!

Table 13: Base template for the experiment of product description generation in this paper.

Method	**Prompt Template**
`Naive`	!<INPUT 4>! = The description should be generated below the "### Generated Result:"
`CoT`	!<INPUT 4>! = Generate your thinking process step by step below the "### Thinking Process:" Then the description should be generated below the "### Generated Result:"
`Plan-N-Solve`	!<INPUT 4>! = Generate your planing step by step below the "### Planning:" Then the description should be generated below the "### Generated Result:"

Table 14: Detail prompt request in Tab. 13.