Delving into RL for Image Generation with CoT: A Study on DPO *vs.* GRPO

Chengzhuo Tong^{*4}, **Ziyu Guo**$^{*\dagger 1}$, **Renrui Zhang**$^{*\ddagger 2}$, **Wenyu Shan**^{*3}, **Xinyu Wei**^{3}
Zhenghao Xing^{1}, **Hongsheng Li**^{2}, **Pheng-Ann Heng**^{1}

CUHK ^{1}IMIXR & ^{2}MMLab ^{3}Peking University ^{4}Shanghai AI Lab

*Equal Contribution †Project Leader ‡Corresponding Author

Abstract

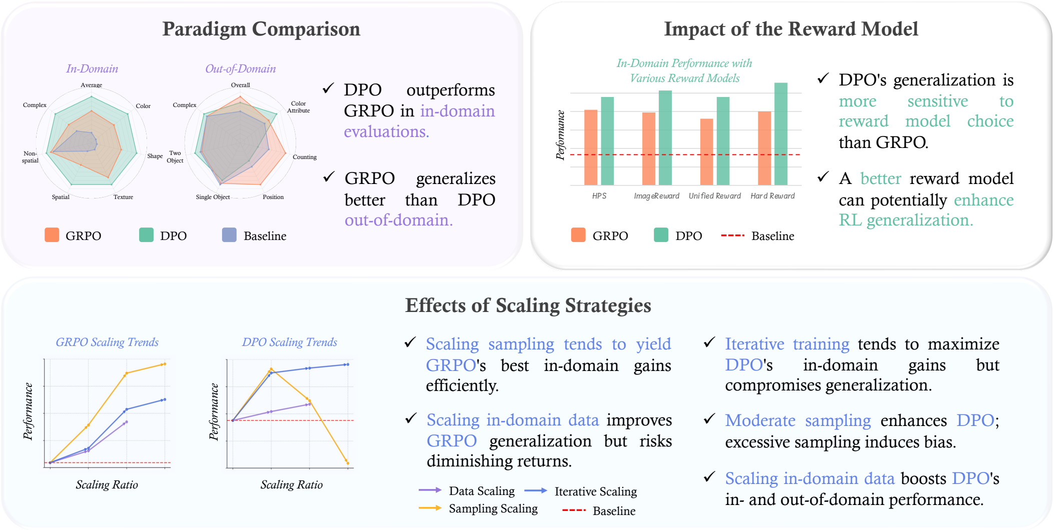
Recent advancements underscore the significant role of Reinforcement Learning (RL) in enhancing the Chain-of-Thought (CoT) reasoning capabilities of large language models (LLMs). Two prominent RL algorithms, Direct Preference Optimization (DPO) and Group Relative Policy Optimization (GRPO), are central to these developments, showcasing different pros and cons. Autoregressive image generation, also interpretable as a sequential CoT reasoning process, presents unique challenges distinct from LLM-based CoT reasoning. These encompass ensuring text-image consistency, improving image aesthetic quality, and designing sophisticated reward models, rather than relying on simpler rule-based rewards. While recent efforts have extended RL to this domain, these explorations typically lack an in-depth analysis of the domain-specific challenges and the characteristics of different RL strategies. To bridge this gap, we provide the first comprehensive investigation of the GRPO and DPO algorithms in autoregressive image generation, evaluating their ***in-domain*** performance and ***out-of-domain*** generalization, while scrutinizing the impact of ***different reward models*** on their respective capabilities. Our findings reveal that GRPO and DPO exhibit distinct advantages, and crucially, that reward models possessing stronger intrinsic generalization capabilities potentially enhance the generalization potential of the applied RL algorithms. Furthermore, we systematically explore ***three prevalent scaling strategies*** to enhance both their in-domain and out-of-domain proficiency, deriving unique insights into efficiently scaling performance for each paradigm. We hope our study paves a new path for inspiring future work on developing more effective RL algorithms to achieve robust CoT reasoning in the realm of autoregressive image generation. Code is released at `https://github.com/ZiyuGuo99/Image-Generation-CoT`.

1 Introduction

Recent large language models (LLMs) [33, 34, 47, 59, 64] have demonstrated remarkable achievements in diverse challenging tasks, such as mathematical problem-solving [4, 19, 32] and code generation [8, 5, 22]. This is driven by their emergent and robust reasoning capabilities through extended Chain-of-Thoughts (CoT) [52, 24, 18, 65, 17, 66], as exemplified by models like OpenAI's o1 [35], DeepSeek-R1 [16], and Kimi k1.5 [45]. These predominant advancements in reasoning abilities are primarily facilitated by reinforcement learning (RL) [3, 61, 30] methods, applied during post-training, which elicits a deliberate, stepwise reasoning process before reaching a final answer.

Among the prominent RL algorithms employed for fine-tuning LLMs, two representative approaches stand out, i.e., Direct Preference Optimization (DPO) [39] and Group Relative Policy Optimization

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



✓ DPO outperforms GRPO in in-domain evaluations.

✓ GRPO generalizes better than DPO out-of-domain.

✓ DPO's generalization is more sensitive to reward model choice than GRPO.

✓ A better reward model can potentially enhance RL generalization.

✓ Scaling sampling tends to yield GRPO's best in-domain gains efficiently.

✓ Scaling in-domain data improves GRPO generalization but risks diminishing returns.

✓ Iterative training tends to maximize DPO's in-domain gains but compromises generalization.

✓ Moderate sampling enhances DPO; excessive sampling induces bias.

✓ Scaling in-domain data boosts DPO's in- and out-of-domain performance.

Figure 1: **Investigation for GRPO and DPO in Autoregressive Image Generation.** We analyze the advantages of GRPO and DPO in both in-domain and out-of-domain scenarios (Top-left), the effect of different reward models (Top-right), and the influence of scaling strategies (Bottom), providing unique insights to this field.

(GRPO) [43]. DPO offers compelling advantages in reduced training time and computational cost, while achieving substantial performance gains, especially in tasks with simpler and shorter CoT reasoning [37, 57]. Nevertheless, DPO's effectiveness diminishes in more complex reasoning scenarios due to its reliance on static, pre-collected data, making it increasingly prone to noise and biases as task complexity rises [57]. In contrast, recent studies underscore GRPO's superior ability to tackle complicated challenges requiring intricate CoT reasoning [16]. It achieves this by iteratively refining policies using self-generated data, effectively adapting to complex task distributions. Nevertheless, GRPO consistently incurs significantly higher computational costs and protracted training durations, as it samples the policy online and continuously during training.

In parallel, image generation [26, 12, 38, 40], one of the most fundamental tasks in multi-modality, also requires substantial reasoning knowledge. Notably, autoregressive generation models [44, 55, 9, 53] can be viewed as a form of CoT reasoning similar to that of LLMs, as they sequentially predict visual tokens in a step-by-step manner. One preliminary work, *'Image Generation with CoT'* [18] has verified the feasibility of DPO in this domain, and follow-up studies [23, 48, 58] experiment with GRPO. However, these endeavors lack in-depth investigation, since the CoT in visual generation exhibits several critical distinctions from textual reasoning. First, image generation tasks often involve prompts describing diverse scenarios, ranging from intricate, detailed descriptions to concise, templated formats. This frequently exhibits a substantial domain gap, with generative models typically excelling in one domain but may struggle in another. Consequently, evaluating the efficacy of RL from both in-domain and out-of-domain perspectives is crucial. Second, distinct from the often verifiable, rule-based rewards of LLMs, image generation objectives, which rely on diverse criteria like text-image alignment and human aesthetic preferences, necessitate a thorough exploration of how different reward models affect the performance of RL algorithms. Third, while recent works, such as [18], have preliminarily adopted scaling strategies like iterative-DPO [37] to improve performance, the impact of many widely used scaling strategies still remains insufficiently explored. Given these challenges, we raise the question: *How does the performance of GRPO compare to that of DPO in autoregressive image generation, and which aspects may influence their optimal performance, e.g., reward models and scaling strategies?*

To this end, we conduct a systematic investigation comparatively evaluating GRPO and DPO regarding their capacity for CoT reasoning in autoregressive image generation. We adopt Janus-Pro [9] as our baseline autoregressive generative model, evaluating its in-domain performance on T2I-CompBench [21] and out-of-domain generalization on GenEval [15]. Specifically, as illustrated in Figure 1, our investigation delves into three primary aspects concerning both GRPO and DPO:

- **In-Domain Performance *vs* Out-of-Domain Generalization.** Examining both in-domain and out-of-domain scenarios enables a comprehensive evaluation of a model's robustness in handling prompts of varying granularity. In our settings, we leverage preference data derived from a unified reward model, conducting a rigorous comparative analysis.

  Observation: *1) The off-policy DPO method demonstrates superior performance on in-domain tasks compared to GRPO. 2) Conversely, GRPO exhibits stronger generalization capabilities, outperforming DPO on the out-of-domain benchmark.*

- **Impact of Different Reward Models.** Reward models define the preference distribution, with diverse models [54, 56, 51, 18] serving this role in text-to-image generation. Understanding how the choice of reward model influences policy outcomes, particularly generalization, is crucial for preference-based methods. To this end, we investigate the differential impact of employing various reward models on the generalization performance of GRPO and DPO, revealing how preference variations shape algorithmic capabilities.

  Observation: *1) DPO exhibits greater sensitivity to reward model variations than GRPO, manifesting larger out-of-domain performance fluctuations. 2) A reward model with superior generalization can potentially improve the generalization performance of RL algorithms.*

- **Investigation of Effective Scaling Strategies.** Investigating how prevalent scaling strategies affect RL's in-domain and out-of-domain performance offers key insights into enhancing model adaptability across tasks. We examine three dimensions: scaling sampled images per prompt, scaling in-domain training data diversity and volume, and adopting an iterative training approach inspired by Iterative-DPO.

  Observation: *GRPO: 1) Scaling sampled images tends to yield more computationally efficient in-domain gains. 2) Moderate scaling of sampling size and in-domain data improve generalization, but excessive scaling risks overfitting. DPO: 1) Scaling iterative training tends to maximize in-domain performance but compromise generalization after multiple cycles. 2) Moderate sampling sharpens preference contrast, optimizing both in-domain and out-of-domain performance, while excessive sampling induces bias. 3) Scaling in-domain data enhances both in-domain and generalization performance by mitigating biases introduced by the limited preference scope of small datasets.*

In summary, our core contributions are as follows:

- We present *the first* comprehensive empirical study comparing GRPO and DPO algorithms for autoregressive image generation, highlighting their respective strengths and providing unique insights into the future advancement of this field.
- By evaluating the inherent generalization capabilities of various reward models and their influence on both GRPO and DPO, we demonstrate that enhancing the generalization capacity of reward models potentially boosts the overall generalization performance of RL.
- We systematically investigate scaling behaviors across multiple critical dimensions, including variations in the number of sampled images per prompt, the scale of the in-domain training data, and the deployment of iterative training paradigms, with the analysis of their differential impact on in-domain performance and out-of-domain generalization, yielding numerous valuable insights and highlighting promising avenues for future research.

2 Our Investigation

Group Relative Policy Optimization (GRPO) [43] and Direct Preference Optimization (DPO) [39] have demonstrated distinct advantages for fine-tuning large language models (LLMs). In this study, following the step of *'Image Generation with CoT'* [18], we conduct a meticulous and systematic investigation aiming to evaluate the efficacy of GRPO and DPO in image generation and identify more impactful strategies for scaling performance.

Table 1: **In-Domain Performance of GRPO and DPO under Various Reward Models.** We assess the performance on the T2I-CompBench benchmark [21], comprehensively evaluating different reward models including HPS [54], ImageReward [56], Unified Reward [51], Fine-tuned ORM [18], and 'Metric Reward'.

Reward Type	Average		Attribute Binding						Object Relationship				Complex	
			Color		Shape		Texture		Spatial		Non-Spatial			
	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO
Baseline	38.56		63.30		34.28		48.90		20.23		30.51		34.12	
HPS	50.49	**53.90**	77.39	85.25	53.59	64.72	71.54	76.08	30.14	25.29	31.10	31.17	39.20	40.89
ImageRwd	49.76	**55.67**	76.57	86.18	52.58	64.92	70.71	76.20	28.82	34.72	31.10	31.53	38.80	40.47
Unified Rwd	48.01	**53.91**	74.34	82.88	50.28	61.74	68.66	76.64	27.80	30.40	30.87	31.25	36.10	40.52
Ft. ORM	53.11	**55.10**	79.49	84.92	56.96	60.87	75.58	76.76	35.97	35.97	31.10	31.26	39.61	40.79
Metric Rwd	50.01	**57.81**	77.21	87.83	53.66	65.91	73.17	79.19	28.69	40.76	30.87	31.53	36.97	41.64

Table 2: **Out-of-Domain Performance of GRPO and DPO under Various Reward Models.** We assessed the generalization performance of both algorithms on the out-of-domain GenEval dataset [15], with reward models aligning consistently with Table 1.

Reward Type	Overall		Color Attr		Counting		Position		Single Object		Two Object		Colors	
	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO
Baseline	78.04		63.50		54.37		76.25		98.44		87.63		88.03	
HPS	**79.18**	77.31	63.00	70.25	60.62	48.12	78.75	67.25	99.69	90.15	86.87	98.75	86.17	89.36
ImageRwd	**79.26**	77.67	59.50	69.25	62.50	43.44	80.50	71.25	98.44	99.69	87.37	90.66	87.23	91.76
Unified Rwd	**80.99**	80.03	67.52	70.54	61.46	55.69	82.02	75.24	99.58	99.00	88.40	91.50	86.96	88.21
Metric Rwd	**79.44**	78.86	65.75	70.29	58.13	51.98	79.50	75.24	98.12	97.46	87.12	89.47	88.03	88.73

2.1 Overview

Task Formulation. To enable the applicability of the prevailing two RL techniques, namely GRPO and DPO, we focus on the autoregressive image generation task, demonstrated by models such as LlamaGen [44], Show-o [55], and Janus-Pro [9]. These models employ a data representation and output paradigm analogous to that of LLMs and large multimodal models (LMMs), while attaining comparable performance to continuous diffusion models [20]. Specifically, they leverage quantized autoencoders [13] to transform images into discrete tokens, enabling the seamless integration of loss mechanisms from both DPO and GRPO during the post-training phase.

Experimental Settings. We select Janus-Pro as our baseline model for this investigation, a latest autoregressive image generation model with advanced capabilities. To comprehensively evaluate the effectiveness of various RL strategies, we assess the text-to-image generation performance on both in-domain and out-of-domain benchmarks. In-domain performance is evaluated using T2I-CompBench [21], which features long, detailed prompts designed to compose complex scenes with multiple objects, displaying various attributes and relationships. Out-of-domain generalization is examined with GenEval [15], utilizing short, templated prompts starting with "a photo of," to assess robustness to concise and standardized textual prompts. In the subsequent sections, we explore three pivotal dimensions to investigate GRPO and DPO for implementing chain-of-thought reasoning in autoregressive image generation: in-domain *vs.* out-of-domain performance (Section 2.2), the impact of reward models (Section 2.3), and the effectiveness of scaling strategies (Section 2.4).

2.2 In-Domain Performance *vs* Out-of-Domain Generalization

Recently, the application of reinforcement learning (RL) has yielded substantial breakthroughs in lifting the reasoning capabilities of LLMs via Chain-of-Thought (CoT) techniques [52, 35, 11, 45, 2, 1]. Existing researches indicate that GRPO, the on-policy RL algorithm, can data-efficiently enhance performance on in-domain (ID) tasks without sacrificing capabilities on out-of-domain (OOD) tasks [28, 31, 14, 29]. Nevertheless, a direct comparison of on-policy GRPO and off-policy Deep Policy Optimization (DPO) regarding their ID and OOD performance under equivalent training data remains underexplored. Drawing inspiration from this, we train and evaluate these two algorithms within the domain of autoregressive image generation to elucidate their comparative strengths.

GRPO. GRPO improves upon Proximal Policy Optimization (PPO) by eliminating the learned value critic and estimating advantages directly through group-wise normalization of rewards across

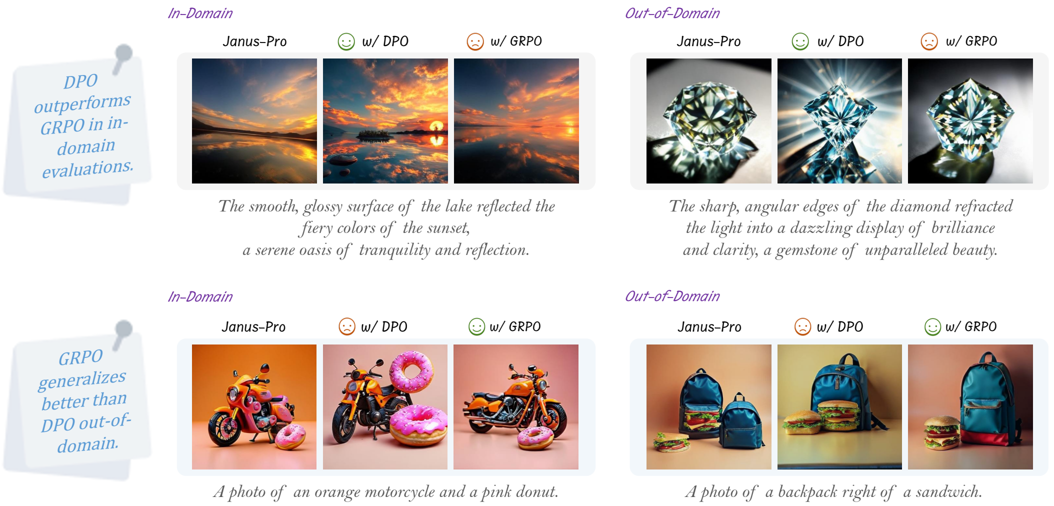

In-Domain
Janus-Pro
☺ w/ DPO
☹ w/ GRPO
DPO outperforms GRPO in in-domain evaluations.
The smooth, glossy surface of the lake reflected the fiery colors of the sunset, a serene oasis of tranquility and reflection.
Out-of-Domain
Janus-Pro
☺ w/ DPO
☹ w/ GRPO
The sharp, angular edges of the diamond refracted the light into a dazzling display of brilliance and clarity, a gemstone of unparalleled beauty.
In-Domain
Janus-Pro
☹ w/ DPO
☺ w/ GRPO
GRPO generalizes better than DPO out-of-domain.
A photo of an orange motorcycle and a pink donut.
Out-of-Domain
Janus-Pro
☹ w/ DPO
☺ w/ GRPO
A photo of a backpack right of a sandwich.

Figure 2: **Visualization Results of In-Domain *vs.* Out-of-Domain Performance Comparison.**

multiple responses per query. Our empirical investigation initiates with a GRPO training phase, employing a group size of four completions per query with the hyperparameter for iteration times set to 1, following the standard GRPO workflow. Building on the design of reward models, as elaborated in 2.3, we focus here on the curation of training data for GRPO. To facilitate a fair comparison between GRPO and DPO, we ensured both methods utilized identical training data. As GRPO's on-policy framework eliminates the need for auxiliary dataset construction, we trained models using the official prompts from the T2I-CompBench dataset, totaling 5.6k prompts.

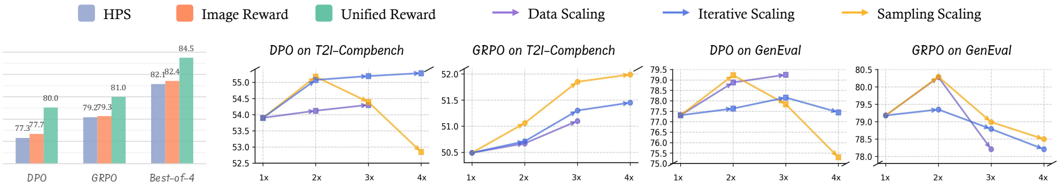
DPO. In contrast to the GRPO on-policy RL paradigm, which solely requires training prompts, DPO functions as an off-policy algorithm, necessitating both prompts and corresponding pairs of chosen and rejected images. Given that autoregressive image generation models are optimized using a cross-entropy loss, we can directly adapt the maximum likelihood objective of DPO to this setting. To clarify, we provide a detailed outline of the DPO methodology from the following two perspectives:

- ***Maintaining Comparable Computational Cost:*** The computational cost of DPO comprises three components: (i) generating training images based on provided prompts, (ii) the scoring of these images by a reward model, and (iii) the subsequent training process. To ensure a fair comparison between DPO and GRPO under comparable computational constraints, we align the number of images generated per prompt in DPO with the group size in GRPO, while employing identical reward models to maintain consistent learning preferences.
- ***DPO Ranking Data Curation:*** Leveraging the images generated by the model, we construct ranking pairs for each prompt by selecting the highest and lowest scoring images as the chosen and rejected images, respectively. This process yields a total of 5.6k ranking pairs for training, derived from the official prompts of the T2I-CompBench dataset.

Experimental Analysis and Insights. As presented in Table 1 and 2, we evaluate the in-domain and out-of-domain performance of GRPO and DPO. We provide qualitative results in Figure 2, with additional visualization in the Supplementary. The key findings are summarized as follows:

- ***DPO demonstrates stronger performance than GRPO in in-domain evaluations.*** As shown in Table 1, DPO consistently outperforms GRPO on T2I-CompBench across various reward models, with DPO's in-domain performance surpassing GRPO by an average of 11.53%. Notably, when using T2I-CompBench's official evaluation tools as the reward signal, DPO attains a peak enhancement of 7.8% over GRPO. This significantly highlights the advantages of DPO over GRPO in terms of in-domain effectiveness and robustness.

- ***GRPO exhibits superior generalization capabilities than DPO in out-of-domain scenarios.*** As illustrated in Table 2, GRPO consistently demonstrates enhanced generalization performance over DPO across various reward models on the GenEval dataset, surpassing DPO by



(a) (b) (c) (d) (e)

Figure 3: **(a) The Impact of Different Reward Models' Intrinsic Generalization Capability.** We evaluate the generalization performance of GRPO, DPO, and the intrinsic generalization performance (represented by best-of-4 strategy) of three reward models. **(b-e) Effects of Three Scaling Strategies.** We examine the effects of various scaling strategies, including sampling size, in-domain data diversity, and iterative training, on both in-domain and out-of-domain performance.

an average of 1.14%. Notably, when HPS serves as the reward model, GRPO achieves a peak improvement of 2.42% over DPO, suggesting its superior generalization capacity.

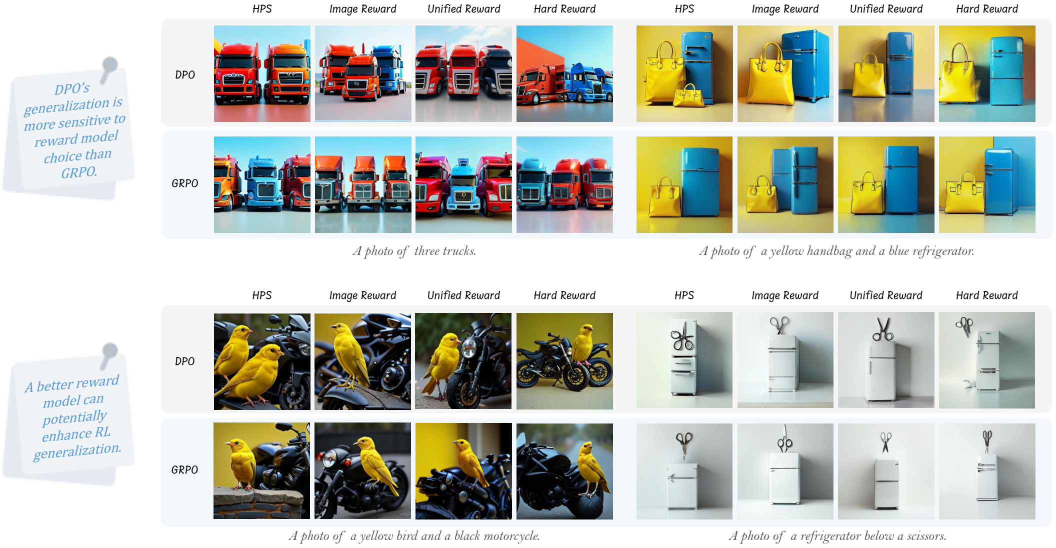
2.3 Impact of Different Reward Models

Recent advancements in LLMs have extensively investigated the influence of reward model variations on performance across diverse tasks, including reasoning [52, 50], safety [36, 67], and general conversation [46, 49]. In text-to-image generation, reward models have been developed to capture human aesthetic and semantic preferences, thereby steering the generative process. However, in autoregressive image generation, the impact of reward model-induced preferences on RL remains limited. A recent study [23] explores the influence of reward models on in-domain RL performance but still lacks exploration of the potential relationship between the intrinsic properties, especially the generalization capabilities, of reward models and RL generalization. To address this gap, we consider exploring the relationship between the generalization of RL and the intrinsic generalization capabilities of reward models, yielding critical insights for the future development of reward models.

Reward Model Design. Unlike domains such as mathematics or programming, where reward signals are typically derived from verification functions that ensure precise alignment with ground-truth solutions, evaluating the quality of generated images necessitates sophisticated learned reward models. The influence of these reward models, along with their inherent biases, on the training process remains underexplored. To elucidate their impact, we examine three distinct types of reward models, each designed to capture unique aspects of image quality and alignment with human preferences:

- ***Human Preference Model.*** Human Preference Models, such as HPS [54] and ImageReward [56], are constructed using vision-language models (VLMs) like CLIP or BLIP to evaluate images based on human aesthetic appeal and text-image alignment. Trained on datasets of human-annotated image rankings, these models provide a holistic assessment of visual quality, generating scores that reflect human-like preferences.
- ***Visual Question Answering Model.*** Visual Question Answering Models, including UnifiedReward [51], Fine-tuned ORM [18] and PARM [18], leverage multimodal large language models (MLLMs) like LLaVA [27] to interpret visual inputs and perform scenario-based evaluations. Trained on diverse datasets comprising images and text, these models emphasize detailed reasoning, enabling precise scoring and evaluation of visual content.
- ***Metric Reward.*** Metric Rewards utilize specialized, domain-specific evaluation tools. In this study, for each training prompt, we identify its corresponding attribute in T2I-CompBench and apply the associated evaluation protocol to score the generated images.

Intrinsic Generalization of Reward Models. In contrast to RL, we adopt Guo et al.'s scalable framework ([18]) for efficiently evaluating reward model (RM) capabilities during inference. This approach uses RMs as outcome reward models (ORMs) in a best-of-N strategy, assessing their capabilities via final scores. We extend this framework to evaluate RM generalization on the GenEval dataset by deploying RMs as ORMs. As shown in Table 3, a best-of-4 selection strategy yields the RM generalization ranking on GenEval: ***Unified Reward > Image Reward > HPS Reward***.



DPO's generalization is more sensitive to reward model choice than GRPO.

A better reward model can potentially enhance RL generalization.

A photo of three trucks.

A photo of a yellow handbag and a blue refrigerator.

A photo of a yellow bird and a black motorcycle.

A photo of a refrigerator below a scissors.

Figure 4: **Visualization Results of the Impact of Different Reward Models.**

Table 3: **Comparison of Reward Model Generalization.** We assess the generalization capabilities of reward models in a best-of-4 selection strategy on the GenEval dataset [15].

Reward Type	Overall	Color Attr.	Counting	Position	Single Obj.	Two Obj.	Colors
HPS	82.14	70.62	77.25	90.15	99.06	87.77	68.00
ImageRwd	82.41	69.69	75.50	93.18	98.75	88.56	68.75
Unified Rwd	84.49	75.00	82.00	92.42	98.75	87.50	71.25

Experimental Analysis and Insights. Our experimental results are visually summarized in Figure 3, and the detailed numeric comparisons are reported in Table 3. We provide qualitative results in Figure 4, with additional visualization in the Supplementary. Then, we draw two principal insights:

- ***DPO's generalization performance exhibits heightened sensitivity to the choice of Reward Model compared to GRPO.*** As presented in Table 2, the range of GRPO's generalization performance on the GenEval dataset is 1.81, whereas DPO's range is notably higher at 2.72. Furthermore, GRPO's performance variance on GenEval is 0.5486, significantly lower than DPO's variance of 0.9547. This suggests that DPO's generalization capabilities are more susceptible to variations in training data preferences, indicating a greater dependency on the specific characteristics of the chosen reward model.

- ***A reward model with superior generalization can potentially improve the generalization performance of RL algorithms.*** As intuitively illustrated in Figure 3 (a), the performance rankings of different reward models on the GenEval dataset, optimized using either GRPO or DPO, remain consistent. Crucially, these rankings align perfectly with our prior evaluations of the intrinsic generalization capabilities of these models. This indicates that the reward model's intrinsic capacity for generalization is a pivotal factor that probably contributes to the overall generalization potential of the RL algorithm.

2.4 Investigation of Effective Scaling Strategies

Prior research has extensively explored methods to optimize the in-domain performance of RL algorithms for LLMs. Notable approaches include iterative DPO (DPO-Iter) [37] and techniques to enhance the efficacy of proximal policy optimization [57]. However, despite these advancements, common scaling behaviors to improve the in-domain (ID) and out-of-domain (OOD) performance of both on-policy and off-policy RL algorithms in autoregressive image generation remain largely underexplored. To address this gap, this section investigates three critical scaling behaviors to enhance the performance of GRPO and DPO across ID and OOD datasets. Specifically, these factors include:

Table 4: **Effect of Scaling Strategies on In-Domain Proficiency.** This table presents the performance evaluation of GRPO and DPO on T2I-CompBench [21] under three distinct scaling strategies: sample scaling (scaling sampled images per prompt), data scaling, and iterative training. Specifically, configurations denoted as 'Base Size' (Data Scaling), 'Sampling 4' (Sample Scaling), and 'Base' (Iterative Training) correspond to the values in Table 1 where the reward type is HPS for comparative analysis. All experiments consistently employ HPS as the reward model.

Scaling Param.	Average		Attribute Binding						Object Relationship				Complex	
			Color		Shape		Texture		Spatial		Non-Spatial			
	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO
Baseline	38.56		63.30		34.28		48.90		20.23		30.51		34.12	
Data Scaling														
Base Size	50.49	53.90	77.39	85.25	53.59	64.72	71.54	76.08	30.14	25.29	31.10	31.17	39.20	40.89
Double Size	50.67	54.12	77.81	82.24	54.39	60.48	72.27	77.06	29.16	33.05	31.15	31.18	39.26	40.70
Triple Size	51.11	54.30	77.73	83.77	56.31	62.35	72.66	77.73	29.21	29.81	31.18	31.28	39.54	40.84
Sample Scaling														
Sampling 4	50.49	53.90	77.39	85.25	53.59	64.72	71.54	76.08	30.14	25.29	31.10	31.17	39.20	40.89
Sampling 8	51.06	55.17	77.79	83.31	55.47	63.86	72.63	75.50	29.99	36.44	31.19	31.05	39.30	40.84
Sampling 12	51.85	54.39	78.18	82.37	56.81	63.17	73.72	77.82	31.51	31.57	31.21	31.14	39.68	40.28
Sampling 16	51.99	52.85	77.38	80.60	59.09	63.49	73.72	75.72	30.70	25.93	31.15	31.06	39.88	40.31
Iterative Training														
Base	50.49	53.90	77.39	85.25	53.59	64.72	71.54	76.08	30.14	25.29	31.10	31.17	39.20	40.89
Iter1	50.71	55.07	77.28	85.14	54.14	64.21	72.35	76.15	29.78	34.05	31.18	31.28	39.51	39.58
Iter2	51.30	55.19	78.02	84.66	55.73	64.44	72.86	76.15	30.76	35.05	31.16	31.29	39.25	39.54
Iter3	51.45	55.28	77.97	84.70	55.94	64.32	73.26	76.32	30.74	35.51	31.19	31.27	39.61	39.53

Table 5: **Effect of Scaling Strategies on Out-of-Domain Generalization.** This evaluation on the GenEval [15] dataset adopts the same scaling strategies as outlined in T2I-CompBench (see Table 4). The configurations, Base Size (Data Scaling), Sampling 4 (Sample Scaling), and Base (Iterative Training), correspond to the values in Table 2, where the reward type is HPS for comparison.

Scaling Param.	Overall		Color Attr		Counting		Position		Single Object		Two Object		Colors	
	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO	GRPO	DPO
Baseline	78.04		63.50		54.37		76.25		98.44		87.63		88.03	
Data Scaling														
Base Size	79.18	77.31	63.00	70.25	60.62	48.12	78.75	67.25	86.87	90.15	99.69	98.75	86.17	89.63
Double Size	80.28	78.88	66.50	67.50	62.81	50.31	78.25	77.00	87.88	91.67	98.75	99.06	87.50	87.88
Triple Size	78.21	79.25	64.50	70.00	56.56	51.25	73.75	76.00	87.12	91.67	98.75	99.06	88.56	87.50
Sample Scaling														
Sampling 4	79.18	77.31	63.00	70.25	60.62	48.12	78.75	67.25	86.87	90.15	99.69	98.75	86.17	89.63
Sampling 8	80.29	79.23	63.25	73.25	63.44	47.81	80.25	70.00	90.15	95.45	96.88	98.75	87.77	90.16
Sampling 12	78.99	77.84	65.25	66.25	60.62	47.81	75.50	70.25	89.14	92.68	97.81	99.38	85.64	90.69
Sampling 16	78.50	75.30	63.50	64.75	57.81	44.69	76.75	63.00	86.62	91.16	99.38	99.38	86.97	88.83
Iterative Training														
Base	79.18	77.31	63.00	70.25	60.62	48.12	78.75	67.25	86.87	90.15	99.69	98.75	86.97	89.36
Iter1	79.35	77.63	67.00	69.00	58.75	54.37	76.50	69.00	87.63	90.91	98.75	98.75	87.50	83.78
Iter2	78.79	78.16	64.75	71.00	57.19	56.62	76.00	66.50	89.14	92.68	98.44	97.81	87.23	85.37
Iter3	78.21	77.45	61.36	65.50	60.63	57.81	75.75	68.75	87.88	91.16	97.50	97.19	86.44	84.31

(1) scaling sampled images per prompt, (2) scaling the diversity of in-domain training data, and (3) implementing iterative training paradigms. The detailed application is delineated as follows:

- ***Scaling Sampled Images per Prompt:*** For GRPO, scaling the quantity of sampled images per prompt corresponds to expanding the group size of real-time samples utilized during the RL training phase. For DPO, where pairs are constructed by selecting the highest- and lowest-scoring images from a pre-generated set for a given prompt, this strategy effectively amplifies the discriminative power of the preference pairs, facilitating more precise alignment with human preferences and the model's ability to distinguish subtle quality differences.

- ***Scaling the Diversity and Quantity of In-Domain Training Data:*** To scale data while rigorously maintaining quality control, we develop a structured prompt generation pipeline leveraging GPT-4o. Building upon the T2I-CompBench, we generated a set of category-specific prompts that was twice as large through carefully constrained API calls (see Supplementary for more implementation details). Our methodology incorporated two key principles:
  1. ***Category-Aware Constraint Preservation:*** All generated prompts maintained strict adherence to their respective category's syntactic templates and semantic boundaries (e.g., 3D spatial prompts required exactly two objects and one spatial relation).

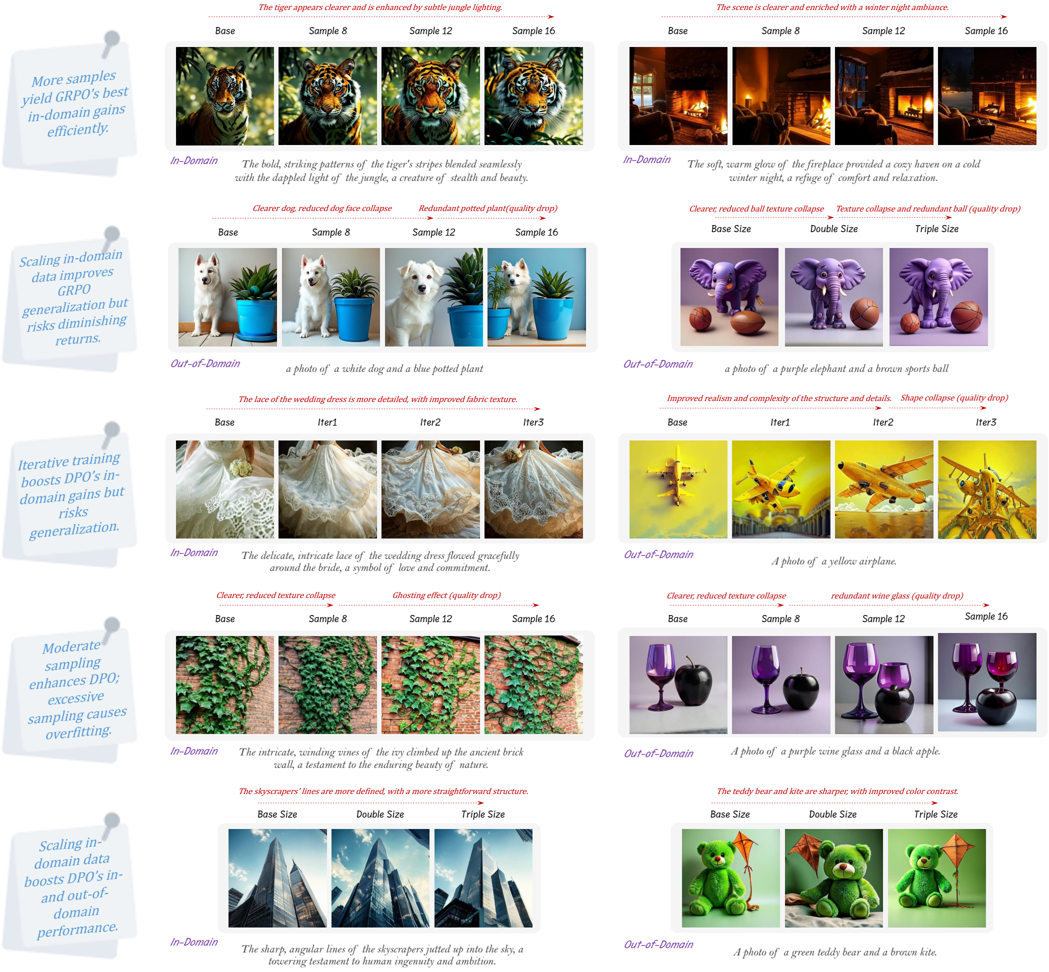

More samples yield GRPO's best in-domain gains efficiently.
The tiger appears clearer and is enhanced by subtle jungle lighting.
Base
Sample 8
Sample 12
Sample 16
In-Domain
The bold, striking patterns of the tiger's stripes blended seamlessly with the dappled light of the jungle, a creature of stealth and beauty.
The scene is clearer and enriched with a winter night ambiance.
Base
Sample 8
Sample 12
Sample 16
In-Domain
The soft, warm glow of the fireplace provided a cozy haven on a cold winter night, a refuge of comfort and relaxation.
Scaling in-domain data improves GRPO generalization but risks diminishing returns.
Clearer dog, reduced dog face collapse
Redundant potted plant(quality drop)
Base
Sample 8
Sample 12
Sample 16
Out-of-Domain
a photo of a white dog and a blue potted plant
Clearer, reduced ball texture collapse
Texture collapse and redundant ball (quality drop)
Base Size
Double Size
Triple Size
Out-of-Domain
a photo of a purple elephant and a brown sports ball
Iterative training boosts DPO's in-domain gains but risks generalization.
The lace of the wedding dress is more detailed, with improved fabric texture.
Base
Iter1
Iter2
Iter3
In-Domain
The delicate, intricate lace of the wedding dress flowed gracefully around the bride, a symbol of love and commitment.
Improved realism and complexity of the structure and details.
Shape collapse (quality drop)
Base
Iter1
Iter2
Iter3
Out-of-Domain
A photo of a yellow airplane.
Moderate sampling enhances DPO; excessive sampling causes overfitting.
Clearer, reduced texture collapse
Ghosting effect (quality drop)
Base
Sample 8
Sample 12
Sample 16
In-Domain
The intricate, winding vines of the ivy climbed up the ancient brick wall, a testament to the enduring beauty of nature.
Clearer, reduced texture collapse
redundant wine glass (quality drop)
Base
Sample 8
Sample 12
Sample 16
Out-of-Domain
A photo of a purple wine glass and a black apple.
Scaling in-domain data boosts DPO's in- and out-of-domain performance.
The skyscrapers' lines are more defined, with a more straightforward structure.
Base Size
Double Size
Triple Size
In-Domain
The sharp, angular lines of the skyscrapers jutted up into the sky, a towering testament to human ingenuity and ambition.
The teddy bear and kite are sharper, with improved color contrast.
Base Size
Double Size
Triple Size
Out-of-Domain
A photo of a green teddy bear and a brown kite.

Figure 5: **Visualization Results of Insights from Investigating Scaling Strategies.**

2. ***Semantic Novelty Enforcement:*** We implemented specific generation constraints to prevent superficial variations, requiring GPT-4o to produce genuinely novel compositions rather than simple lexical substitutions. This yields semantically distinct yet plausible object pairings (e.g., *"A person is exploring a forest and taking photos of the wildlife"* ***vs.*** original *"A child is playing with a toy airplane in the backyard"*), significantly expanding the conceptual coverage of our training data.

- ***Implementing Iterative Training Paradigms:*** The motivation for adopting iterative training paradigms lies in their ability to progressively enhance model performance by leveraging updated reference policies, thereby reducing overfitting risks and improving generalization. Inspired by this, we develop iterative variants of GRPO and DPO, termed GRPO-Iter and DPO-Iter, respectively. For GRPO-Iter, we extend the standard GRPO framework with iterative cycles, updating the reference model with the policy model's parameters after each training round, which ensures the KL penalty aligns with the current policy. For DPO-Iter, we followed iterative DPO protocols[37, 57], where preference pairs are re-sampled and re-evaluated after each training cycle, with the reference model updated accordingly.

Experimental Analysis and Insights. As shown in Table 4 and 5, we conduct a comprehensive investigation into the efficacy of three key scaling strategies. To enable a fair comparison of GRPO and DPO, we standardize the experimental setting by adopting the HPS [54] as the reward model to compare the effectiveness of the three scaling strategies, with additional visualizations provided in the Supplementary due to space limitations. Our principal findings are summarized as follows:

- ***Scaling sampled images per prompt tends to yield more computationally efficient in-domain gains for GRPO:*** Qualitative visualizations in Figure 3 (c) for GRPO on T2I-CompBench demonstrate that scaling group size in autoregressive image generation outperforms both in-domain data scaling and iterative scaling. This enhanced performance is driven by improved advantage estimation with larger group sizes, which stabilizes policy updates and strengthens in-domain optimization. Given that doubling each factor (group size, in-domain data scale, or iterative training paradigms) incurs approximately comparable computational cost, prioritizing scaling sampling emerges as a highly efficient strategy.

- ***Moderately scaling the sample size and increasing the diversity and quantity of in-domain data improves generalization for GRPO, but excessive scaling diminishes growth:*** In autoregressive image generation, moderate sampling size within the GRPO algorithm (e.g., to 4 or 8) and in-domain training data quantity (e.g., by doubling or tripling) progressively improves generalization, as shown in Figure 3 (e) for GRPO on GenEval. However, when larger sample sizes are employed or when GRPO's in-domain data scale is *triple*, generalization exhibits a slight decline due to overfitting to in-domain characteristics, which underscores the critical need to balance in-domain optimization with robust generalization.

- ***Iterative training tends to maximize DPO's in-domain performance but risks generalization degradation:*** Iterative training substantially enhances the in-domain performance of DPO for autoregressive image generation, as evidenced by the Iterative Scaling curve in Figure 3 (b) and (d). Notably, a single iteration of DPO-Iter outperforms the results obtained by tripling the training data, with additional iterations providing further incremental gains in in-domain metrics. However, generalization on the GenEval declines significantly after two iterations, likely due to overfitting the training preference data, underscoring the trade-off between maximizing preference alignment and maintaining robust generalization.

- ***Moderate sampling optimizes DPO's preference contrast for improved in-domain and generalization performance, while excessive sampling induces bias:*** In autoregressive image generation, scaling the sample size for DPO preference pair selection yields non-monotonic performance effects, as illustrated by the DPO sampling scaling curve in Figure 3 (b) and (d). Relative to the baseline, sample sizes of 4, 8, and 16 yield in-domain gains of 37.38%, 40.61%, and 34.71%, alongside out-of-domain generalization changes of -0.93%, $+1.52\%$, and -3.51%, respectively. This suggests that scaling the sample size optimizes preference contrast while avoiding biases introduced by excessive scaling.

- ***Scaling in-domain data for DPO optimizes performance across both in-domain and out-of-domain by mitigating preference bias:*** Scaling in-domain data by factors of one, two, and three achieves relative in-domain improvements of 37.40%, 37.97%, and 38.37% over the baseline, respectively. While single-scale training results in a 0.94% decline on the out-of-domain benchmark, scaling by factors of two and three produces gains of 1.08% and 1.55%. This highlights that carefully curating a diverse and representative set of preference pairs is critical to overcoming the constrained preference scope inherent in small datasets, thereby mitigating potential in domain and out-of-domain performance degradation.

3 Conclusion

In this paper, we conducted a rigorous experimental analysis demonstrating that DPO excels in in-domain tasks, while GRPO exhibits superior out-of-domain generalization. We further establish that the generalization capacity of reward models potentially shapes both algorithms' generalization potential. Through systematic exploration of three scaling strategies, we derive critical insights for achieving enhanced Chain-of-Thought reasoning in autoregressive image generation.

Acknowledgement

The work described in this paper was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project 14201321 and Project 14200824.

Overview of Appendix

- Appendix A: Related work.
- Appendix B: Implementation details of structured prompt generation pipeline.
- Appendix C: Detailed record of computational time.

A Related Work

Visual Generative Models. Visual generative models have advanced through two primary paradigms: autoregressive and diffusion approaches. Autoregressive methods, inspired by language modeling success [33, 34, 47, 59], sequentially predict image tokens or pixels, as seen in ViT-VQGAN [60] and VideoPoet [25]. Recent work like LlamaGen [44] demonstrates that pure autoregressive architectures can achieve state-of-the-art generation, while Janus [9] introduces decoupled visual encoding to unify multimodal understanding and generation. Meanwhile, diffusion models have emerged as a powerful alternative, with continuous approaches [63] dominating text-to-image tasks and discrete variants like MaskGIT [7] operating on tokenized representations. Notably, Show-o adopts discrete diffusion through masked token prediction, achieving high-fidelity generation while maintaining training efficiency.

Reinforcement Learning (RL). Reinforcement Learning (RL) trains agents to maximize rewards through environment interactions, with methods split into on-policy (e.g., PPO [42], GRPO [43]) and off-policy (e.g., DPO [39]) approaches. On-policy methods like PPO use current policy data for stable but costly updates, employing techniques like GAE [41] for variance reduction, while GRPO replaces critics with group-wise reward comparisons. Off-policy methods like DPO reuse historical data for efficiency but risk distribution mismatch, directly optimizing preferences without reward modeling. The key difference lies in data usage: on-policy requires fresh data for stability, whereas off-policy trades some reliability for sample efficiency. Applied to language models via RLHF [6], these methods enhance alignment (e.g., RLOO [3]'s critic-free approach) and reasoning [10, 16, 45, 62] through MDP formulations, balancing computational cost and performance in tasks like mathematical reasoning. This demonstrates RL's adaptability across policy paradigms for improving language models.

B Implementation Details of Structured Prompt Generation Pipeline

As discussed in Sec. 2.4 of the main paper (*Investigation of Effective Scaling Strategies*), we enlarge T2I-CompBench by generating an additional set of category-specific prompts with GPT-4o, thereby **doubling** the size of the original benchmark. Specifically, for each of the eight categories, *color*, *texture*, *shape*, *numeracy*, *spatial*, *3D spatial*, *non-spatial*, and *complex*, we craft a dedicated *meta-prompt*. All meta-prompts are derived from a shared template, but include category-dependent constraints. An example for the *color* category is given after the following paragraph.

In practice, we iterate through every prompt in the *color* subset of T2I-CompBench, replace the placeholder `#Prompts From T2I-CompBench#` with the current prompt, and feed the resulting meta-prompt to GPT-4o. We apply the same pipeline to the remaining seven categories. The complete collection of category-specific meta-prompts will be released upon the paper's acceptance.

Color Category Example

```
I am working on a reinforcement learning for image generation project, and I
need your assistance in generating additional prompts that focus on color-based
descriptions.
Existing prompts: #Prompts From T2I-CompBench#
Task: Generate 2 additional prompts that maintain the same syntactic structure
while ensuring diversity.
Requirements:
Color Usage:
Each prompt must explicitly include at least two different colors.
The color words should be commonly used and perceptually distinct (e.g., "red"
and "blue" are good, but "light red" and "dark red" are too similar).
Allowed color descriptors: basic colors (e.g., red, blue, green, yellow, pink,
purple, orange, brown, black, white, gray) and common material-based variations
(e.g., "golden", "silver", "ivory").
Avoid uncommon or overly specific colors (e.g., "cerulean", "chartreuse").
Object Selection:
The first object should be a tangible item with a strong association to color
(e.g., clothing, furniture, makeup, vehicles, buildings).
The second object (if applicable) should also be a realistic, color-relevant
entity that fits within a scene.
Avoid repetition of objects already in the dataset (e.g., if "lipstick" and
"blush" exist, do not use them again).
Color and Object Compatibility:
Ensure that the selected colors are realistically applicable to the given
objects.
Examples of Good Color-object Pairings:
"A red sports car and a black leather seat." (both colors are reasonable for cars
and seats)
Examples of Bad Color-object Pairings (to avoid):
"A purple banana and a silver cloud." (unnatural color choices)
Diversity Constraints:
Do not generate prompts that are simple color swaps (e.g., "A red lipstick and a
pink blush" and "a pink lipstick and a red blush" are too similar).
Ensure semantic diversity by describing different types of objects and settings
(e.g., fashion, interior design, nature, technology).
The sentence structure should mimic the provided examples but not be identical.
Output Format:
Return the response as a Python list of strings in JSON-compatible format, e.g.:
{
"prompt1",
"prompt2"
}
Strictly use lowercase (no capitalization except for proper nouns).
Now, generate two new prompts following these requirements.
```

Table 6: **Comparison of DPO and GRPO Training Computational Costs (in GPU hours).**

Reward Type	**DPO**				**GRPO**
	Simple Image	*Scoring*	*Training*	***Total***	***Total***
HPS	1.51 h	0.83 h	0.67 h	2.99 h	2.92 h
ImageReward	1.50 h	0.08 h	0.67 h	2.25 h	2.55 h
Unified Reward	1.51 h	1.80 h	0.67 h	3.97 h	4.03 h

C Detailed Record of Computational Time

To facilitate a fair comparison between DPO and GRPO, as outlined in Sections 2.2, we maintain comparable training computational costs, measured in terms of computational time. The computational expense of DPO consists of three main components: (i) generating training images based on provided prompts, (ii) scoring these images using a reward model, and (iii) executing the subsequent training phase. Detailed computational times for both GRPO and DPO are systematically recorded and presented in Table 6. Additionally, we assess and document the total training computational time for three key scaling strategies implemented for GRPO and DPO across different scaling ratios, as presented in Table 7. These computational time costs for both tables are evaluated using 8 A100 GPUs, with Janus-Pro [9] serving as the baseline.

Table 7: **Total Computational Time for Scaling Strategies Across Varying Scaling Ratios.**

Scaling Strategy	***Ratio 1***		***Ratio 2***		***Ratio 3***	
	DPO	**GRPO**	**DPO**	**GRPO**	**DPO**	**GRPO**
Data Scaling	2.99 h	2.92 h	5.97 h	5.84 h	9.01 h	8.76 h
Sampling Scaling	2.99 h	2.92 h	5.33 h	5.78 h	7.66 h	8.64 h
Iterative Scaling	2.99 h	2.92 h	5.99 h	5.84 h	8.98 h	8.76 h

References

[1] Claude 3.7 sonnet. https://www.anthropic.com/claude/sonnet/, 2025.

[2] Gemini 2.5 pro. https://deepmind.google/technologies/gemini/pro/, 2025.

[3] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

[4] Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*, 2019.

[5] Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. *CoRR*, abs/2108.07732, 2021.

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

[7] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[8] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021.

[9] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-Pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

[10] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

[11] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[12] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[13] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021.

[14] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. Video-R1: Reinforcing video reasoning in mllms, 2025.

[15] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

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

[17] Ziyu Guo, Ray Zhang, Hao Chen, Jialin Gao, Dongzhi Jiang, Jiaze Wang, and Pheng-Ann Heng. Sciverse: Unveiling the knowledge comprehension and visual reasoning of lmms on multi-modal scientific problems. *arXiv preprint arXiv:2503.10627*, 2025.

[18] Ziyu Guo, Renrui Zhang, Chengzhuo Tong, Zhizheng Zhao, Peng Gao, Hongsheng Li, and Pheng-Ann Heng. Can we generate images with cot? let's verify and reinforce image generation step by step. *arXiv preprint arXiv:2501.13926*, 2025.

[19] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

[20] Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models, 2022.

[21] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2I-CompBench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.

[22] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *CoRR*, abs/2403.07974, 2024.

[23] Dongzhi Jiang, Ziyu Guo, Renrui Zhang, Zhuofan Zong, Hao Li, Le Zhuo, Shilin Yan, Pheng-Ann Heng, and Hongsheng Li. T2I-R1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*, 2025.

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

[25] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. VideoPoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

[26] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[27] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[28] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. VideoChat-R1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

[29] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. VideoChat-R1: Enhancing spatio-temporal perception via reinforcement fine-tuning, 2025.

[30] Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*, 2025.

[31] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-RFT: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025.

[32] MAA. American invitational mathematics examination - aime. In *American Invitational Mathematics Examination - AIME 2024*, February 2024.

[33] OpenAI. Chatgpt. https://chat.openai.com, 2023.

[34] OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024.

[35] OpenAI, Aaron Jaech, Adam Kalai, Adam Lerer, et al. OpenAI o1 system card, 2024.

[36] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[37] Richard Yuanzhe Pang, Weizhe Yuan, He He, Kyunghyun Cho, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *Advances in Neural Information Processing Systems*, 37:116617–116637, 2024.

[38] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[39] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

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

[41] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

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

[44] Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

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

[46] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, et al. LaMDA: Language models for dialog applications, 2022.

[47] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[48] Junke Wang, Zhi Tian, Xun Wang, Xinyu Zhang, Weilin Huang, Zuxuan Wu, and Yu-Gang Jiang. SimpleAR: Pushing the frontier of autoregressive visual generation through pretraining, sft, and rl. *arXiv preprint arXiv:2504.11455*, 2025.

[49] Xiaolong Wang, Yile Wang, Yuanchi Zhang, Fuwen Luo, Peng Li, Maosong Sun, and Yang Liu. Reasoning in conversation: Solving subjective tasks through dialogue simulation for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

[50] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[51] Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

[52] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[53] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024.

[54] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

[55] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

[56] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

[57] Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024.

[58] Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

[59] 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. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[60] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.

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

[62] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772, 2024.

[63] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023.

[64] Renrui Zhang, Jiaming Han, Chris Liu, Aojun Zhou, Pan Lu, Yu Qiao, Hongsheng Li, and Peng Gao. Llama-adapter: Efficient fine-tuning of large language models with zero-initialized attention. In *ICLR 2024*, 2024.

[65] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, et al. MathVerse: Does your multi-modal llm truly see the diagrams in visual math problems? *ECCV 2024*, 2024.

[66] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, et al. Mavis: Mathematical visual instruction tuning with an automatic data engine. *arXiv preprint arXiv:2407.08739*, 2024.

[67] Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. SafetyBench: Evaluating the safety of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims made in the abstract and introduction accurately reflect this paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We discuss the limitations of the work in Supplementary.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please see Section Experiments, and the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We use publicly available datasets and will provide open access to the code upon acceptance.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please see the Investigation section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We conduct experiments only once and report the accuracy of the best model, and it would be too computationally expensive to conduct the pre-training multiple times.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please see Supplementary.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: This work focuses on academic, publicly-available datasets including T2I-CompBench and GenEval. This work is not related to any private or personal data, and there's no explicit negative social impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Please see Section Investigation and Supplementary.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.