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

Anonymous Author(s)

Affiliation

Address

email

1 Overview of Appendix

- 2 • Appendix **A**: Related work.
- 3 • Appendix **B**: Implementation details of structured prompt generation pipeline.
- 4 • Appendix **C**: Detailed record of computational time.
- 5 • Appendix **D**: Additional qualitative results.
- 6 • Appendix **E**: Limitations and future work

7 A Related Work

8 **Visual Generative Models.** Visual generative models have advanced through two primary
9 paradigms: autoregressive and diffusion approaches. Autoregressive methods, inspired by lan-
10 guage modeling success [8, 9, 16, 17], sequentially predict image tokens or pixels, as seen in
11 ViT-VQGAN [18] and VideoPoet [7]. Recent work like LlamaGen [14] demonstrates that pure
12 autoregressive architectures can achieve state-of-the-art generation, while Janus [4] introduces decou-
13 pled visual encoding to unify multimodal understanding and generation. Meanwhile, diffusion models
14 have emerged as a powerful alternative, with continuous approaches [20] dominating text-to-image
15 tasks and discrete variants like MaskGIT [3] operating on tokenized representations. Notably, Show-o
16 adopts discrete diffusion through masked token prediction, achieving high-fidelity generation while
17 maintaining training efficiency.

18 **Reinforcement Learning (RL).** Reinforcement Learning (RL) trains agents to maximize rewards
19 through environment interactions, with methods split into on-policy (e.g., PPO [12], GRPO [13])
20 and off-policy (e.g., DPO [10]) approaches. On-policy methods like PPO use current policy data for
21 stable but costly updates, employing techniques like GAE [11] for variance reduction, while GRPO
22 replaces critics with group-wise reward comparisons. Off-policy methods like DPO reuse historical
23 data for efficiency but risk distribution mismatch, directly optimizing preferences without reward
24 modeling. The key difference lies in data usage: on-policy requires fresh data for stability, whereas
25 off-policy trades some reliability for sample efficiency. Applied to language models via RLHF [2],
26 these methods enhance alignment (e.g., RLOO [1]’s critic-free approach) and reasoning [5, 6, 15, 19]
27 through MDP formulations, balancing computational cost and performance in tasks like mathematical
28 reasoning. This demonstrates RL’s adaptability across policy paradigms for improving language
29 models.

30 B Implementation Details of Structured Prompt Generation Pipeline

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

```
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.
```

37

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

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 1. 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 2. These computational time costs for both tables are evaluated using 8 A100 GPUs, with Janus-pro [4] serving as the baseline.

Table 1: 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

51

Table 2: 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

D Additional Qualitative Results

We provide additional visualization results for “In-Domain Performance vs. Out-of-Domain Generalization” and “the Impact of Different Reward Models” in Figure 1 and Figure 2. We also provide more qualitative results for the “Effect of Scaling Strategies” in Figure 3. These examples further support and illustrate the key insights discussed in the main paper.

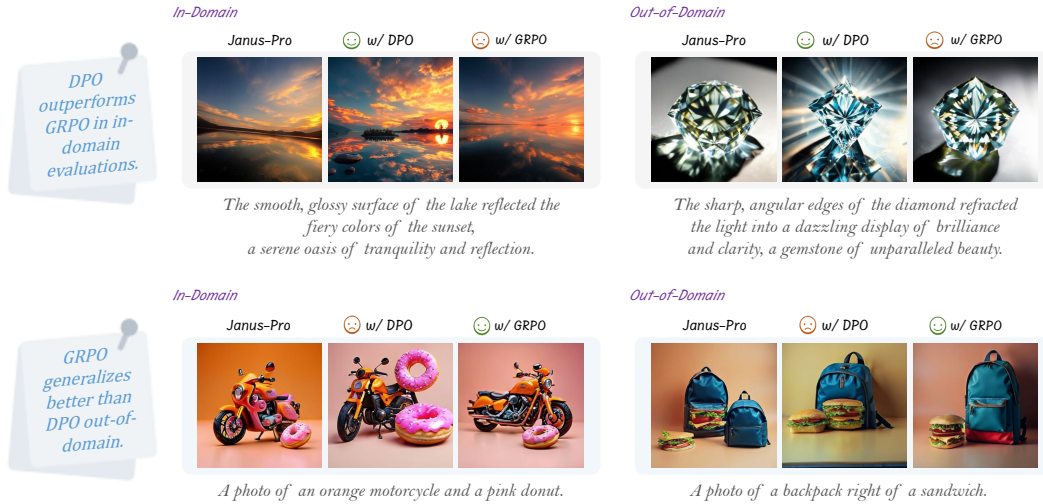


Figure 1: More Visualization Results of In-Domain vs Out-of-Domain Performance Comparison.

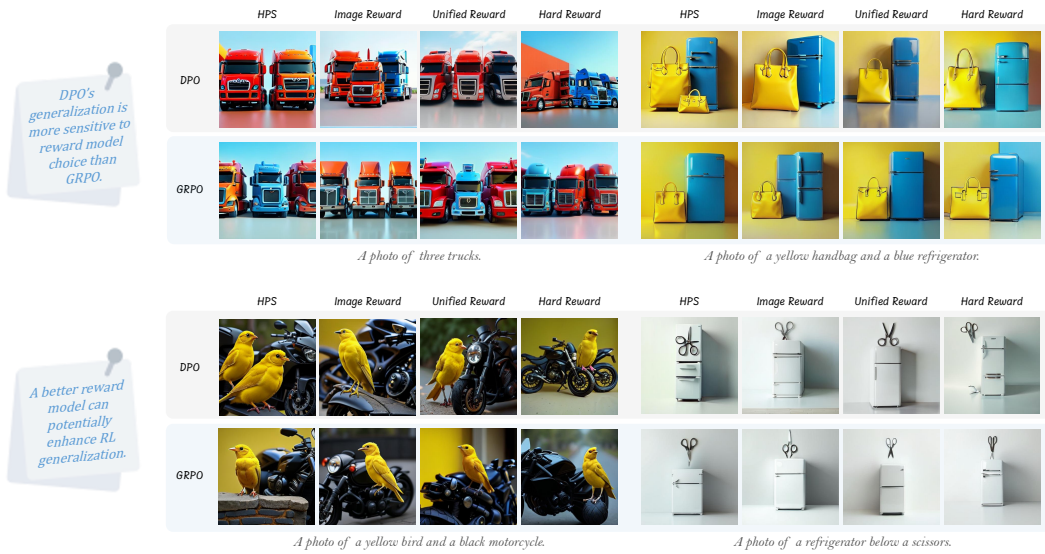


Figure 2: More Visualization Results of the Impact of Different Reward Models.

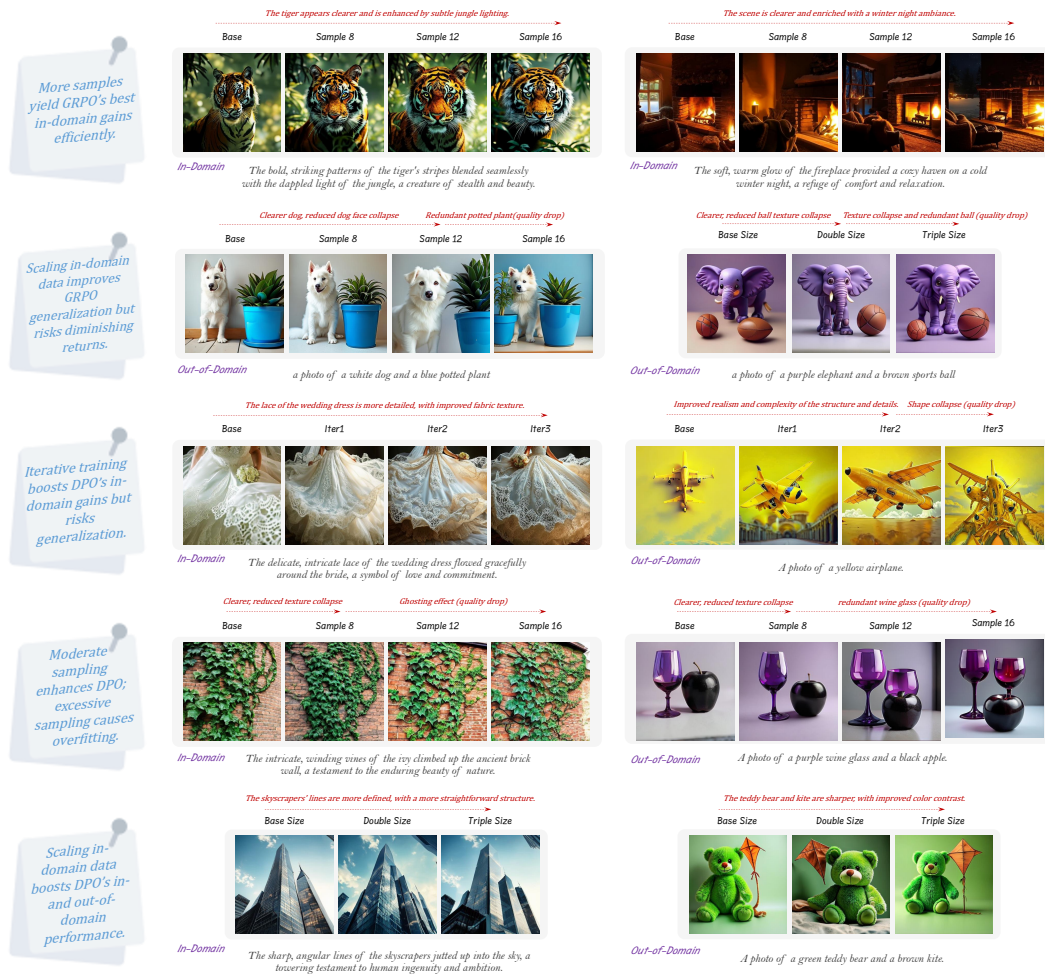


Figure 3: Visualization Results of Different Investigating Scaling Strategies.

E Limitations and Future Work

While our study establishes a strong foundation for incorporating Chain-of-Thought reasoning into image generation via reinforcement learning (RL), several aspects warrant further investigation. Our current focus is on two representative RL algorithms and autoregressive generation frameworks, leaving broader explorations, such as alternative RL strategies, diffusion-based paradigms, and extensions to other generative tasks like video or 3D content, as valuable future directions. These areas offer opportunities to deepen understanding and enhance the generalizability of RL-based approaches in multimodal generation.

References

- [1] 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.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] OpenAI. Chatgpt. <https://chat.openai.com>, 2023.
- [9] OpenAI. Hello gpt-4o. <https://openai.com/index/hello-gpt-4o/>, 2024.
- [10] 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.
- [11] 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.
- [12] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [13] 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.
- [14] 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.

- 104 [15] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li,
105 Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement
106 learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.
- 107 [16] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timo-
108 thée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open
109 and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- 110 [17] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li,
111 Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong
112 Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou,
113 Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li,
114 Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie
115 Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong
116 Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan,
117 Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and
118 Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.
- 119 [18] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku,
120 Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with
121 improved vqgan. *arXiv preprint arXiv:2110.04627*, 2021.
- 122 [19] Dan Zhang, Sining Zhou, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*:
123 Llm self-training via process reward guided tree search. *Advances in Neural Information*
124 *Processing Systems*, 37:64735–64772, 2024.
- 125 [20] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image
126 diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*,
127 pages 3836–3847, 2023.