GRPO-Guard: Mitigating Implicit Over-Optimization in Flow Matching via Regulated Clipping

Supplementary Material

Overview This supplementary material provides additional details to support the main paper:

1. Full version of the Related Work section, including extended discussions and citations.
2. Detailed descriptions of training procedures, hyperparameters, and evaluation protocols.
3. Human evaluation results across various settings and tasks.
4. Additional analyses of GRPO-Guard, including the relationship between reward hacking and denoising steps, as well as statistics on clipping fraction.
5. Comparison with KL Regularization.

1. Related Works

1.1. Alignment for Large Language Models

Recent years witness a shift from supervised fine-tuning to interactive, reinforcement-style alignment when adapting Large Language Models(LLMs) [1] to human intent [26, 29]. Reinforcement Learning from Human Feedback (RLHF) [9] — which typically trains a reward model from pairwise human comparisons and then optimizes a policy using RL algorithms such as PPO [25] — becomes a standard pipeline for this purpose [2, 5, 22]. However, PPO-based RLHF pipelines are often computationally intensive and sensitive to reward-model inaccuracies, which has motivated the development of more stable and efficient alternatives. One such direction is Direct Preference Optimization (DPO) [23], which bypasses explicit reinforcement learning by directly optimizing model parameters on human preference pairs, achieving similar alignment effects with reduced complexity. More recently, Group Relative Policy Optimization(GRPO) methods have already been adopted in production-scale LLM alignment flows [11, 14], demonstrating that group-relative updates can yield stable improvements in instruction following and preference alignment.

1.2. RL for Diffusion and Flow Models.

Diffusion and flow-matching models [13, 17, 24, 28] decompose the process of visual generation into iterative denoising steps, revolutionizing the field of visual synthesis and achieving remarkable results in both image and video generation. Building on the success of reinforcement learning (RL) algorithms in Large Language Models (LLMs), similar optimization paradigms—such as PPO [3, 25] and DPO [30]—have been effectively transferred to diffusion models, enabling preference alignment and improved task-specific performance. Following this trend, Flow-GRPO [18] and DanceGRPO [36] integrate GRPO-style policy updates into flow-matching models, transforming deterministic ODE sampling into stochastic SDE formulations to introduce exploration noise for group-based optimization. More recently, MixGRPO [16] proposes a hybrid ODE–SDE sampling strategy that significantly improves training efficiency while maintaining comparable generation quality. Meanwhile, Flow-CPS [31] identifies a critical issue in the SDE sampling process used by FlowGRPO and DanceGRPO—namely, the inconsistency of noise coefficients across timesteps—which leads to excessive residual noise and inaccurate reward estimation. To address this, Flow-CPS introduces a noise-consistent SDE sampling scheme that accelerates GRPO optimization by improving reward accuracy. In parallel, TempFlowGRPO [12] and G^2RPO [37] address the reward sparsity and inaccuracy caused by assigning a single global reward to multi-step SDE trajectories. Most existing methods focus on improving policy optimization efficiency but overlook a critical issue—over-optimization, which severely degrades visual quality. In this work, we conduct an in-depth analysis of this problem and propose an effective solution.

1.3. Reward Over-optimization.

Reward over-optimization [9, 21], also referred to as reward hacking [20, 27], poses a significant challenge in reinforcement learning for diffusion and flow models, arising from the limitations of imperfect proxy reward models [19, 32, 35] (RMs) for human or task-specific preferences. In practice, optimizing a learned proxy RM often improves its corresponding proxy metric, but alignment with the true objective—such as perceptual quality or human-evaluated preference—typically holds only for a short period, after which further optimization can degrade generation quality, as illustrated in Figure **??**.

To mitigate this issue, common strategies include regularizing policy updates with a heavy KL-divergence penalty [8, 18] toward a supervised fine-tuned policy. KL regularization helps mitigate over-optimization by reducing drift from the reference policy, but it can also slow the improvement of both proxy scores and true-performance metrics, potentially leading to degraded overall performance. Clipping importance ratios [25] further constrains updates from overly confident positive and negative samples, preventing harmful updates and stabilizing policy op-

timization, thereby reducing the risk of entering an over-optimization phase. Additionally, scaling up reward models [9, 33], using ensembles [6, 7], or composing RMs from multiple perspectives can further reduce overfitting to a single proxy, although at significant computational cost. Early stopping [3] and monitoring generation quality provide additional safeguards against excessive reward exploitation, but they may also halt training prematurely, potentially leaving the policy under-optimized.

However, in flow-matching models, the inherent bias in the importance ratio causes the clipping mechanism to fail to function as intended, allowing overly confident positive updates to pass unchecked and driving the policy into an over-optimization regime. In this work, we analyze this phenomenon in depth and propose methods to mitigate implicit over-optimization, thereby restoring stable and reliable policy updates.

2. Experimental Setting

Implementation Details: We conduct experiments on two baselines, Flow-GRPO [18] and DanceGRPO [36], using two backbone models, SD3.5-M and Flux.1-dev, to validate the effectiveness of our method in mitigating reward hacking. Following the Flow-GRPO setting, we apply LoRA fine-tuning for both baselines, with the LoRA rank set to 32, the scaling factor α set to 64, a learning rate of 3e-4, and a clip range of 1e-4. For GRPO-Guard, due to the differences in ratio distributions and gradient magnitudes across steps, we set the clip range to 2e-6, with a learning rate of 1e-4 on SD3.5-M and 2e-4 on Flux.1-dev. Notably, since PickScore rewards exhibit relatively minor reward hacking, we use a smaller clip range of 4e-6. All experiments are conducted on 16 × NVIDIA A800 GPUs. KL loss is not applied. The training and validation datasets are kept consistent with FlowGRPO.

Evaluation Metrics: Following Flow-GRPO, we conduct experiments on three proxy tasks: GenEval [10], TextRender [4], and PickScore [15]. GenEval is a rule-based evaluation framework that assesses a generator's ability to follow textual instructions by measuring object count, color consistency, and spatial arrangement. PickScore is derived from human preference data, where a regression head is fine-tuned on a CLIP encoder so that its scores align with human judgments. To comprehensively evaluate reward hacking, we further construct a composite gold score based solely on **image quality**, measured by HPSv2 [34], ImageReward [35], and UnifiedReward [32]. During training, we monitor the gold score online by using PickScore for the GenEval and TextRender tasks. For the validation datasets, GenEval, PickScore, and TextRender use the corresponding validation sets from FlowGRPO, while HPSv2, ImageReward, and UnifiedReward all use the PickScore validation set.

3. Human Evaluation

We conduct a human preference evaluation to assess image quality, text alignment, and overall quality between the baseline methods and GRPO-Guard. On both the Geneval and OCR tasks, human evaluators compare 100 sample pairs, and the win/tie/lose ratios are shown in Figure 1. The results demonstrate a clear superiority of GRPO-Guard in both image quality and overall quality, indicating that the baseline methods suffer from severe over-optimization, leading to a notable degradation in visual fidelity.

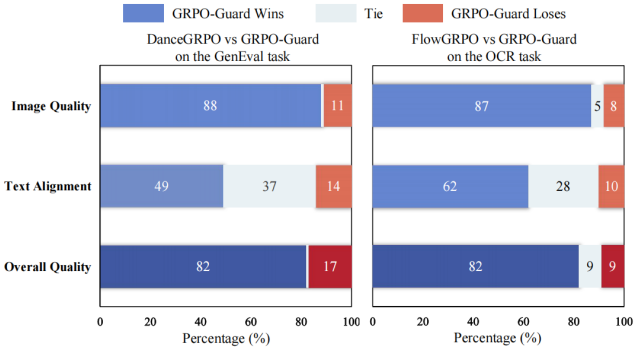

Figure 1. Human evaluation results.

4. Analysis

Hacking Step: Due to the malfunctioning clipping mechanism, gradients from all steps with importance ratios exceeding $1+\epsilon$ are not truncated. Consequently, the hacking model exhibits abnormal behaviors across all denoising stages. we visualize the one-step sampled x_0 predictions from v_θ at different diffusion steps, as shown in Figure 2. **At high-noise steps**, the hacking model shows clear pathological patterns: the generated images contain overly simplistic and uniform structures—typically limited to the main subjects such as a dog and a table—while omitting broader contextual elements. The global layout appears to be determined prematurely, leaving little room for diverse or detailed scene composition. **At low-noise steps**, compared with the base model, the hacking model loses its ability to refine fine-grained details. Even during the final denoising stages, substantial residual noise and artifacts remain, resulting in degraded visual quality. These observations indicate that the hacking model suffers from persistent capability degradation throughout the entire denoising process, which aligns with our analysis that gradients beyond $1+\epsilon$ are never clipped across all timesteps—ultimately causing severe over-optimization.

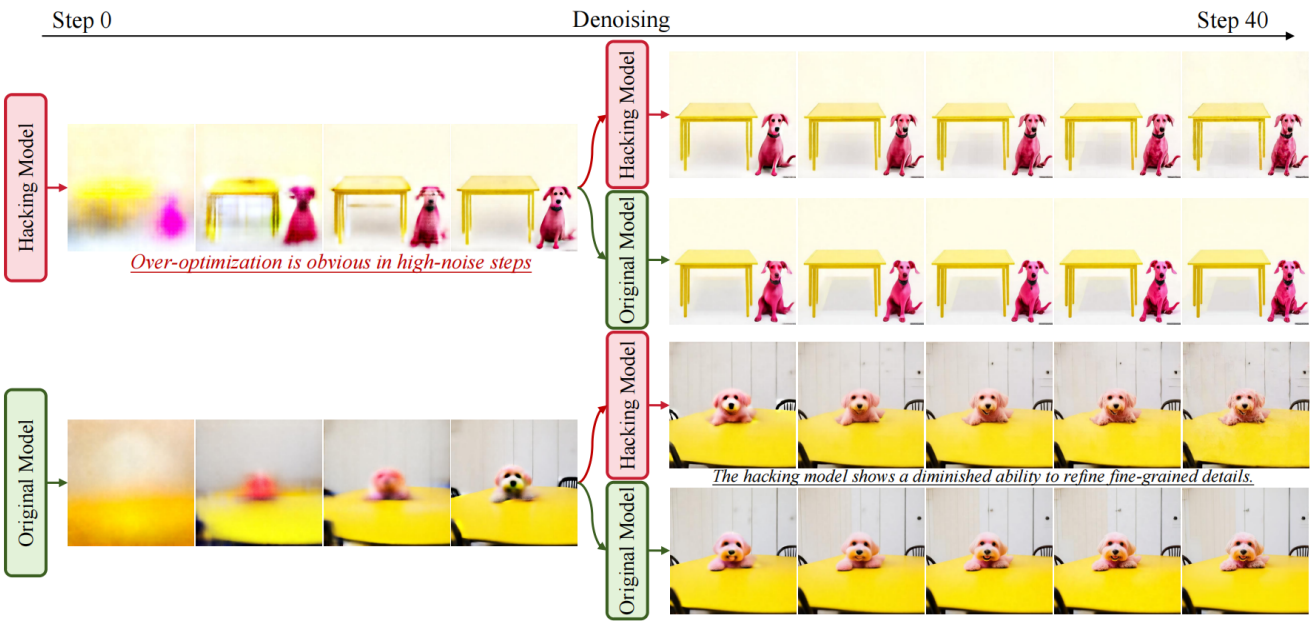

Figure 2. Performance differences between the hacking model and the original model across different denoising steps.

Clip Fraction: We statistically analyze and visualize the clipping ratios of the baseline methods FlowGRPO and GRPO-Guard across different denoising steps. The proportions of samples with importance ratios $r(\theta)$ larger than $1+\epsilon$and smaller than $1-\epsilon$ are recorded separately, as shown in the Figure 4. As expected, in FlowGRPO, a large number of clipping events with ratios smaller than $1-\epsilon$ occur only at the final step (step 8), while the proportion of clipping with ratios larger than $1+\epsilon$ — corresponding to truncation of gradients with positive advantages — remains zero. This imbalance leads to the over-optimization phenomenon. In contrast, GRPO-Guard exhibits more stable and balanced clipping ratios across all steps, with the proportions of $>1+\epsilon$ and $<1-\epsilon$ clipping remaining roughly equal. This indicates that the distributional bias of the ratio has been effectively corrected and the unhealthy clipping mechanism has been mitigated.

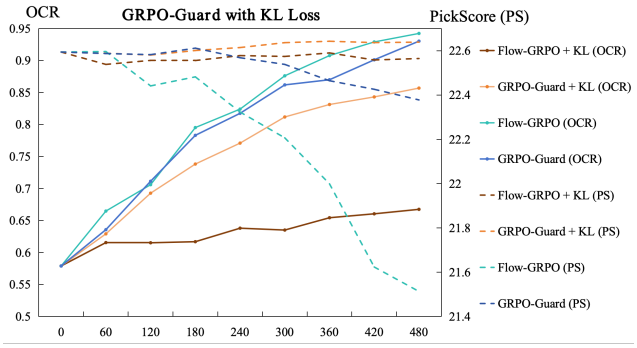

Figure 3. Comparison with KL Regularization.

5. Comparison with KL Regularization

Following your suggestion, we conduct experiments on the OCR task with four settings (all with a KL coefficient of 0.04): 1) GRPO-Guard + KL, 2) Flow-GRPO + KL, 3) Flow-GRPO, 4) GRPO-Guard. As shown in Figure 3, incorporating standard KL regularization into Flow-GRPO substantially reduces optimization efficiency, consistent with prior observations. In contrast, GRPO-Guard achieves significantly faster and more stable optimization, and attains a higher Gold Score even when combined with KL regularization. Notably, GRPO-Guard + KL consistently outperforms KL-regularized Flow-GRPO, demonstrating that the proposed method remains effective in practical, KL-regularized settings. This difference can be attributed to the fundamentally different roles of the two mechanisms. Standard KL regularization mitigates reward hacking by globally constraining the policy to remain close to the reference model, acting as a persistent hard constraint throughout optimization, which inevitably suppresses learning efficiency. In contrast, GRPO-Guard provides a targeted and predictive safeguarding mechanism: it dynamically identifies and truncates potentially harmful updates based on real-time distributional shifts, while leaving benign updates largely unaffected.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al.

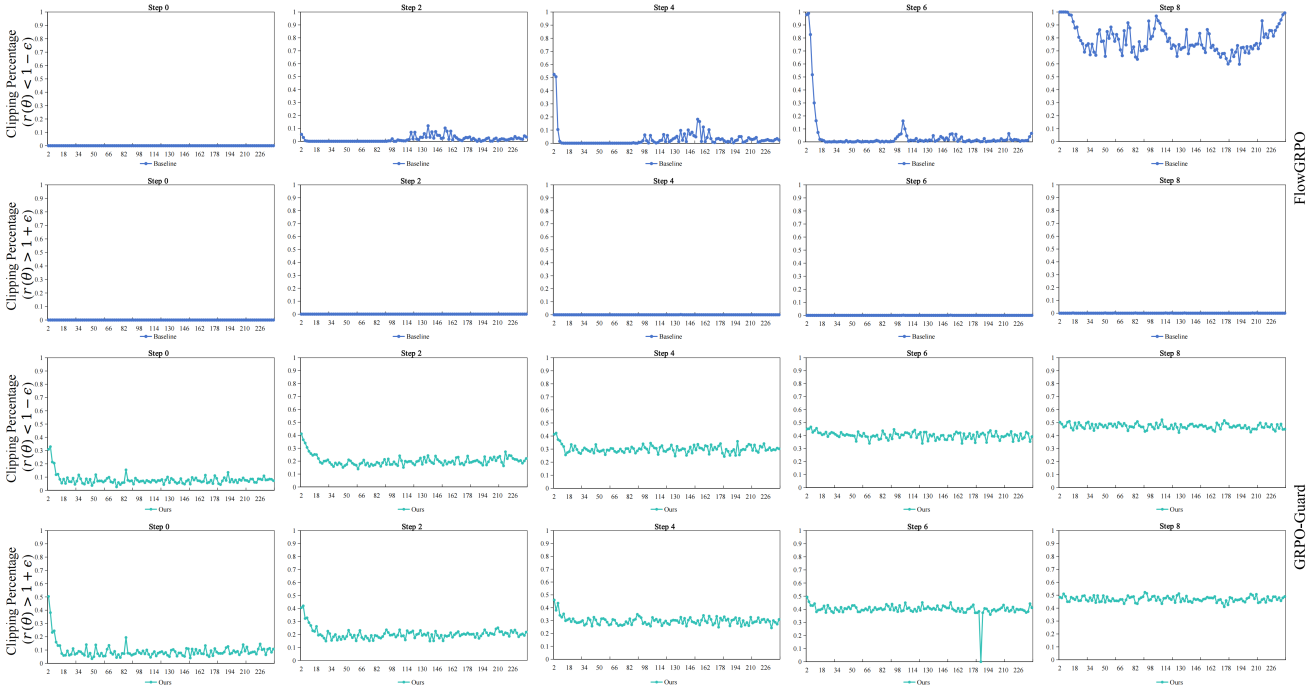

Figure 4. Clipping percentage of $r(\theta) < 1 - \epsilon$ and $r(\theta) > 1 + \epsilon$ during training for FlowGRPO and GRPO-Guard across different denoising steps.

Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1

[3] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023. 1, 2

[4] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *Advances in Neural Information Processing Systems*, 36:9353–9387, 2023. 2

[5] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017. 1

[6] Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*, 2023. 2

[7] Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D'Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023. 2

[8] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36:79858–79885, 2023. 1

[9] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023. 1, 2

[10] 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. 2

[11] 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. 1

[12] Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. Tempflow-grpo: When timing matters for grpo in flow models. *arXiv preprint arXiv:2508.04324*, 2025. 1

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[14] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. 1

[15] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Ma-

tiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023. 2

[16] Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Miles Yang, and Zhao Zhong. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*, 2025. 1

[17] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 1

[18] Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*, 2025. 1, 2

[19] Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Wenyu Qin, Menghan Xia, et al. Improving video generation with human feedback. *arXiv preprint arXiv:2501.13918*, 2025. 1

[20] Yuchun Miao, Sen Zhang, Liang Ding, Rong Bao, Lefei Zhang, and Dacheng Tao. Inform: Mitigating reward hacking in rlhf via information-theoretic reward modeling. *Advances in Neural Information Processing Systems*, 37: 134387–134429, 2024. 1

[21] Ted Moskovitz, Aaditya K Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca D Dragan, and Stephen McAleer. Confronting reward model overoptimization with constrained rlhf. *arXiv preprint arXiv:2310.04373*, 2023. 1

[22] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 1

[23] 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. 1

[24] 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. 1

[25] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 1

[26] Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, Idan Szpektor, et al. Multi-turn reinforcement learning with preference human feedback. *Advances in Neural Information Processing Systems*, 37:118953–118993, 2024. 1

[27] Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022. 1

[28] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1

[29] Hao Sun. Supervised fine-tuning as inverse reinforcement learning. *arXiv preprint arXiv:2403.12017*, 2024. 1

[30] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024. 1

[31] Feng Wang and Zihao Yu. Coefficients-preserving sampling for reinforcement learning with flow matching. *arXiv preprint arXiv:2509.05952*, 2025. 1

[32] 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. 1, 2

[33] Jie Wu, Yu Gao, Zilyu Ye, Ming Li, Liang Li, Hanzhong Guo, Jie Liu, Zeyue Xue, Xiaoxia Hou, Wei Liu, et al. Rewarddance: Reward scaling in visual generation. *arXiv preprint arXiv:2509.08826*, 2025. 2

[34] 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. 2

[35] 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. 1, 2

[36] 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. 1, 2

[37] Yujie Zhou, Pengyang Ling, Jiazi Bu, Yibin Wang, Yuhang Zang, Jiaqi Wang, Li Niu, and Guangtao Zhai. G2rpo: Granular grpo for precise reward in flow models. *arXiv preprint arXiv:2510.01982*, 2025. 1