Smart-GRPO: Smartly Sampling Noise for Efficient RL of Flow-Matching Models

Benjamin Yu[*1,2], **Ziyang Liu**[*2], **Justin Cui**[1],

[1]University of California, Los Angeles,
[2]Brown University,
{yubenjamin2022, justincui}@ucla.edu, ziyang_liu@brown.edu

Abstract

Recent advances in flow-matching models have enabled high-quality text-to-image generation. However, the deterministic nature of flow-matching makes these models poorly suited to reinforcement learning, a key paradigm for improving image quality and aligning outputs with human preferences. Prior work introduces stochasticity by perturbing latents with noise sampled uniformly at random, but most such perturbations yield low-reward generations and lead to inefficient, unstable optimization. We propose Smart-GRPO, the first method to optimize noise perturbations for reinforcement learning in flow-matching models. Smart-GRPO employs an iterative search strategy that decodes candidate perturbations, evaluates them with a reward model, and progressively refines the noise distribution toward higher-reward regions. Experiments show that Smart-GRPO improves both reward optimization and visual quality over baseline methods under comparable compute budgets. These results highlight a practical path toward reinforcement learning in flow-matching frameworks, narrowing the gap between efficient training and human-aligned generation.

Introduction

Flow-matching models (Lipman et al. 2023) have recently been introduced as an alternative to diffusion-based generative models (Ho, Jain, and Abbeel 2020), offering more stable training and deterministic sampling. While large-scale pre-training enables these models to produce high-quality outputs, it is often insufficient for ensuring alignment with human preferences. Reinforcement learning with human feedback (RLHF) (Ouyang et al. 2022), originally developed for aligning large language models, has since been adapted to generative vision models, including diffusion architectures (Black et al. 2023; Yang et al. 2024).

Extending RL to flow-matching models, however, presents distinct challenges. The deterministic nature of flow-matching sampling is fundamentally misaligned with the stochasticity required for policy optimization. Flow-GRPO (Liu et al. 2025) addresses this by introducing random perturbations to the inputs prior to denoising, thereby enabling the use of Group Relative Policy Optimization (GRPO) (Shao et al. 2024). While this modification permits reinforcement learning, it is intrinsically inefficient: most noise seeds sampled uniformly at random produce low-reward generations that contribute little to policy improvement, resulting in wasted training signal. This observation motivates a more principled treatment of noise selection as a key factor in improving the efficiency of RL for flow-matching models.

Intuitively, different noise seeds induce different trajectories through the flow model's latent space: some seeds reliably lead to generations whose rewards are highly sensitive to policy updates, while others produce low-reward, low-informative samples. Treating all seeds as equally valuable forces the learner to spend a significant portion of its optimization budget on trajectories that convey little gradient information about human preferences. This perspective suggests that the choice of noise is not merely a technical detail, but an important lever for shaping the effectiveness of reinforcement learning on flow-matching models.

Prior works have focused primarily on improving training stability (Wang et al. 2025; Xue et al. 2025), efficiency through optimization strategies (Li et al. 2025a,b), or generative fidelity (He et al. 2025). Our work specifically focuses on the noise used to perturb the inputs. We hypothesize that, by directly optimizing the sampling of input noise, we provide a complementary pathway for improving both efficiency and alignment in reinforcement learning for flow-based generative models.

In this work, we introduce **Smart-GRPO**, a framework that augments Flow-GRPO with reward-guided noise selection. Our central hypothesis is that noise seeds vary in their contribution to effective learning, and that preferentially sampling informative seeds can accelerate convergence. Smart-GRPO employs a pretrained reward model to evaluate candidate noise seeds and selects those predicted to yield higher-quality generations. This procedure can be viewed as constructing an adaptive curriculum over the noise distribution, where training gradually emphasizes seeds that produce more informative trajectories. By iteratively refining the noise distribution in this manner, Smart-GRPO improves the efficiency of policy optimization while maintaining compatibility with existing RLHF pipelines.

[*]These authors contributed equally.

Related Works

Flow-matching models: Let $x_0 \in X_0$ be a sample from a true distribution and let $x_1 \in X_1$ be a sample from a known distribution (e.g. a Gaussian). Flow-matching models (Esser et al. 2024) define a path between the data and the noise as a linear interpolation:

$$x_t = (1-t)x_0 + tx_1, \quad t \in [0,1]. \tag{1}$$

Taking the derivative with respect to t yields the target velocity field:

$$\frac{dx_t}{dt} = x_1 - x_0. \tag{2}$$

The goal is then to learn a parameterized velocity predictor $v_\theta(x_t, t)$ that approximates this ground-truth field. This is achieved by minimizing the following loss (Lipman et al. 2023):

$$\mathbb{L}(\theta) = \mathbb{E}_{t,x_0,x_1}\left[||(x_1 - x_0) - v_\theta(x_t,t)||^2\right]. \tag{3}$$

Compared to diffusion (Ho, Jain, and Abbeel 2020) models, which learn a score function or directly predict noise, flow-matching instead learns the velocity of the probability flow ODE. This provides a more direct parameterization of the generative process, and in practice can lead to faster and more stable training.

Reinforcement Learning for Generative Models: Reinforcement learning from human feedback (RLHF) (Ouyang et al. 2022) has become a standard approach for aligning large language models with human preferences. The framework typically involves first training a reward model on collected preference data, and then fine-tuning the language model using Proximal Policy Optimization (PPO) (Schulman et al. 2017). More recently, alternatives such as Direct Preference Optimization (DPO) (Rafailov et al. 2023) and Group Relative Policy Optimization (GRPO) (Shao et al. 2024) have been proposed, offering simpler and more flexible formulations of preference-based training.

Recently, reinforcement learning has also been adapted to diffusion models, which present unique challenges due to their iterative denoising process. Approaches such as Denoising Diffusion Policy Optimization (DDPO) (Wallace et al. 2024) and Direct Preference for Denoising Diffusion Policy Optimization (D3PO) (Yang et al. 2024) extend preference-based optimization to the diffusion setting, enabling alignment with human or other task-specific objectives.

Reinforcement Learning for Flow-matching models: Due to the deterministic nature of flow-matching models, they are not intrinsically designed for reinforcement learning. The probability flow ODE deterministically maps inputs to outputs, leaving little room for the stochastic exploration that reinforcement learning requires. This mismatch makes direct application of standard policy optimization methods ineffective.

Flow-GRPO (Liu et al. 2025) addresses this by converting the deterministic probability flow ODE into an equivalent stochastic differential equation (ODE-to-SDE), which injects randomness while preserving the model's marginal distributions, and by introducing a denoising reduction strategy that reduces the number of denoising steps during training while keeping the full schedule at inference. These modifications enable the incorporation of GRPO into flow-matching models. Empirically, Flow-GRPO achieves substantial gains in compositional image generation, text rendering, and human preference alignment, while maintaining image quality and minimizing reward hacking.

Methods

We introduce Smart-GRPO, an efficient algorithm for fine-tuning flow-matching models with reinforcement learning. Our method improves upon GRPO-style approaches by directly searching over the noise variables that determine the decoded output. Instead of perturbing latents with random noise (as in GRPO), Smart-GRPO searches for noise that maximizes reward in one-shot decoding. We treat the noise distribution as a parameterized search space. Instead of blindly perturbing latents, we iteratively refine a Gaussian noise distribution toward regions of higher reward using a Cross-Entropy Method (CEM)-like update.

Algorithm

Let X_t denote the latent at timestep t, and let $f : X \to \mathbb{R}$ be a scalar reward function. Smart-GRPO proceeds as follows: We first initialize a Gaussian distribution over noise variables, parameterized by mean $\mu = 0$ and standard deviation $\sigma = I$. Importantly, we assume independence, so as to prevent costly covariance calculations. In each iteration, we sample K candidate noises as:

$$m_i = \mu + \sigma n_i, n_i \sim N(0, I) \tag{4}$$

We then perturb the latent with the noise via the following equation:

$$Z_i = X_t + \sqrt{-dt}\sigma_t m_i \tag{5}$$

where σ_t is the noise scale and dt is the step size. To evaluate the effect of each perturbation, we form a one-step approximation of the decoded image using the predicted velocity v_θ.

$$x_0^{(i)} \approx z_i - t v_\theta(z_i, t) \tag{6}$$

This is intended to provide a rough estimate of the final image without requiring the full reverse process. Note that at earlier timesteps (high noise levels), the one-step approximation produces near-random outputs, making reward evaluation unreliable. Smart-GRPO is therefore most effective at later timesteps where the latent has a stronger correlation with the decoded image

We then decode the image, and calculate the reward $R_i = f(x_0^{(i)})$. We then select the top $T = \lfloor P \cdot K \rfloor$, where $P \in [0,1]$ candidates with the highest rewards, using these noises to update the μ and σ used to sample. This process is repeated N times.

Algorithm 1: Smart-GRPO

Require: Latent image X_t, number of sampled noises K, number of iterations N, saving fraction $P \in [0, 1]$, reward function $f(z) : X \to \mathbb{R}$

Ensure: Optimized latent mean μ or sampled latent $m = \mu + \sigma \cdot n$

1: Initialize $\mu = 0$, $\sigma = I$ of the shape of latent variable
2: **for** $n = 1$ **to** N **do**
3: Sample K random noises $\{n_i\}_{i=1}^{K}$ and compute modified noises $m_i = \mu + \sigma \cdot n_i$
4: Perturb the latent with noise:

$$Z_i = X + \sqrt{-dt} \cdot \sigma_t \cdot m_i$$

5: Decode latents Z_i from m_i and compute reward $R_i = f(Z_i)$
6: Select top $T = \lfloor P \cdot K \rfloor$ noises with highest rewards
7: Update mean and standard deviation:

$$\mu = \frac{1}{T}\sum_{i=1}^{T} m_i, \quad \sigma^2 = \frac{1}{T}\sum_{i=1}^{T}(m_i - \mu)^2$$

8: **end for**
9: **return** μ or $m = \mu + \sigma n$

This update step shifts the distribution toward higher-reward regions while adaptively controlling its spread, ensuring a balance between exploration and exploitation.

Once this process is complete, either the mean noise μ or a final sample $m = \mu + \sigma n$ is drawn to be used to perturb the latent for training.

Experiments

Figure 1: Example generations from SD 3.5M from the (i) base model, the (ii) model fine-tuned using FlowGRPO and (iii) the model fine-tuned with SmartGRPO.

Experiments

This section describes the experimental setup used to empirically evaluate whether Smart-GRPO improves the performance of flow-matching models. We compare against strong pretrained and RL-based baselines, optimize two complementary reward functions, and train and evaluate all methods under carefully matched conditions.

Baselines

To assess the effectiveness of our algorithm for fine-tuning flow-matching models, we consider three pretrained base models and one RL baseline:

- **Stable Diffusion 3.5-M (SD 3.5M)** (Esser et al. 2024), the medium-sized rectified-flow model that serves as our primary backbone.
- **Stable Diffusion 3.5-L (SD 3.5L)** (Esser et al. 2024), a larger-capacity variant used as a stronger pretrained reference model.
- **FLUX.1-dev** (Batifol et al. 2025), a recent flow-matching model designed for high-quality in-context image generation and editing.
- **SD 3.5M + Flow-GRPO** (Liu et al. 2025), where the SD 3.5M backbone is fine-tuned using Flow-GRPO without our noise-selection algorithm.

Smart-GRPO is applied on top of the SD 3.5M backbone, and we compare against both the pretrained base models and the Flow-GRPO fine-tuned model.

Reward Functions

We optimize and evaluate Smart-GRPO using two reward models that capture complementary aspects of text-to-image quality:

- **ImageReward**, a general-purpose reward model trained to jointly assess prompt–image alignment, visual fidelity, and harmlessness. It therefore serves as a broad proxy for overall generation quality.
- **Aesthetic Score**, a model that directly targets visual appeal and style, reflecting how pleasing an image is to human perception.

Using both rewards allows us to evaluate Smart-GRPO under different alignment objectives. ImageReward emphasizes semantic consistency and robustness, whereas Aesthetic Score focuses on stylistic quality. In our experiments, we consider both settings where a single reward is optimized (either ImageReward or Aesthetic Score) and report performance on both metrics.

Prompt Datasets

For RL fine-tuning, we follow the setup of Flow-GRPO and build on its GenEval-based prompt pool (Ghosh, Hajishirzi, and Schmidt 2023). Concretely, we randomly sample a training corpus of 3,000 prompts from the datasets provided in the Flow-GRPO repository. We use 2,700 prompts for training and reserve 300 prompts as a held-out validation set for monitoring reward curves and checking for overfitting during training.

For the main quantitative comparison reported in Table 1, we additionally evaluate all models on an independent set of 1,000 prompts sampled from the same GenEval-based pool. This evaluation set is fixed across methods and is used only for offline comparison of ImageReward and Aesthetic Score.[1]

[1]In all tables and plots, we report mean scores across all prompts in the evaluation set.

Training Protocol

We initialize all RL runs from the publicly released Stability AI Stable Diffusion 3.5-Medium checkpoint and reuse most hyperparameters from the original Flow-GRPO implementation for a fair comparison.[2] All experiments are conducted on a single NVIDIA H100 GPU.

During training, we generate images at a resolution of 512×512 using 10 sampling steps with a classifier-free guidance scale of 4.5. Each training batch contains 4 images per prompt, yielding an effective batch size of 4. We set the number of batches per outer epoch to 8 and configure gradient accumulation such that two parameter updates occur per epoch. We include a KL regularization term with weight $\beta = 0.04$ and maintain an exponential moving average (EMA) of model weights to stabilize training and evaluation. Model checkpoints are saved every 60 epochs, at which point we also run evaluation on the held-out prompts.

For Smart-GRPO, we set the number of refinement iterations N and the number of candidate noises per iteration K such that the total number of decoded candidates and reward model evaluations is comparable to that of Flow-GRPO. This ensures that any performance differences between the two methods cannot be attributed simply to using more compute. Unless otherwise specified, we use $N = 5$ and $K = 5$, with a saving fraction P that retains the top-scoring subset of candidates for updating the mean and variance of the noise distribution.

Evaluation Setup

At evaluation time, we use 40 sampling steps and keep the classifier-free guidance scale fixed to 4.5 for all methods to ensure comparability. For each prompt in the evaluation set, we generate one image per model and compute both ImageReward and Aesthetic Score. When optimizing a single reward (e.g., ImageReward), we report the optimized reward as the primary metric, and the auxiliary reward (e.g., Aesthetic Score) as a secondary metric to examine potential trade-offs.

In addition to aggregate quantitative metrics (Table 1), we inspect qualitative generations across a range of prompts. Figure 1 illustrates representative samples from the SD 3.5M base model, SD 3.5M fine-tuned with Flow-GRPO, and SD 3.5M fine-tuned with Smart-GRPO. This combination of quantitative and qualitative evaluation allows us to assess not only whether Smart-GRPO yields higher rewards, but also whether the resulting images are visually coherent, diverse, and aligned with the input prompts.

Analysis

For both rewards we experimented on, our method has both better performance and more stable compared to base Flow-GRPO. Figure 2 shows that Smart-GRPO consistently improves ImageReward scores across training epochs, converging faster and achieving higher final reward than Flow-GRPO. Over our evaluation dataset, Smart-GRPO consistently outperforms the baseline models, as shown in Table 1

[2]See the appendix for the full list of hyperparameters.

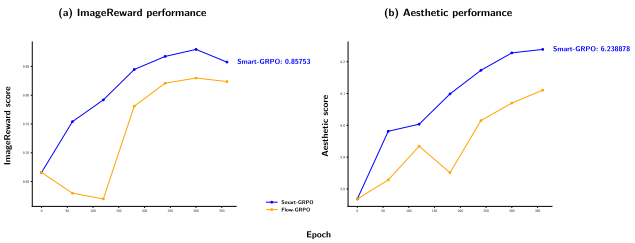

Figure 2: Training results of Smart-GRPO over 360 epochs. Figure **(a)** is trained with ImageReward, and Figure **(b)** is trained using the Aesthetic score

Ablation Study

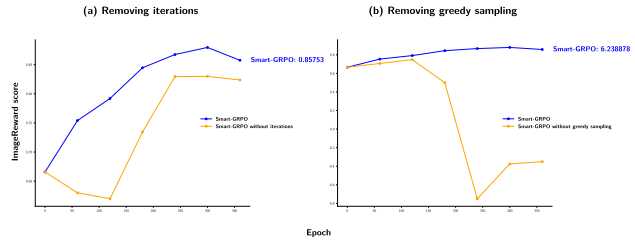

Figure 3: Training results for ablation studies. Figure (a) compares the performance of the algorithm in model performance when when we do or do not include iterations. Figure (b) compared the performance of the algorithm in model performance when when we do or do not include greedy sampling.

To better understand the contribution of Smart-GRPO's core mechanisms, we conduct ablation studies on its two central components: **iterative refinement** and **greedy noise selection**. All ablations were performed under the same training hyperparameters and with the ImageReward reward function.

Iterative refinement. Instead of progressively updating the noise distribution, we evaluate a one-shot alternative in which 25 noise samples are drawn, the top 12 are selected, and their mean is used to perturb the latent. We compare this baseline to Smart-GRPO with $N = 5$ iterations and $K = 5$ sampled noises per iteration, so that both methods consume a comparable total number of forward passes. As shown in Figure 3a, both approaches achieve similar performance, but Smart-GRPO consistently outperforms the one-shot baseline across training steps. This suggests that iterative refinement enables the model to repeatedly concentrate its sampling distribution around higher-quality noise regions, gradually filtering out suboptimal directions that would otherwise be averaged in a single-shot update. In practice, this leads to a tighter, more targeted exploration of the noise space and yields stronger overall performance for roughly the same compute budget.

Greedy noise selection. To assess the role of greedy selection, we replace high-reward noise selection with random sampling, keeping $N = 5$ and $K = 5$. As shown in

Model	ImageReward	Aesthetic Score
Stable Diffusion 3.5M	0.6658	5.769
Stable Diffusion 3.5L	-0.0310	5.602
FLUX.1-dev	0.5121	6.093
SD 3.5M (with Flow-GRPO)	0.8237	6.111
SD 3.5M (with Smart-GRPO)	0.8575	6.238

Table 1: Smart-GRPO model results over evaluation dataset. Dataset consists of 1000 sample prompts generated from GenEval scripts, provided in Flow-GRPO's repository. Results are means over scores of generated images from prompt dataset.

Figure 3b, this variant exhibits highly unstable training: repeatedly updating with low-quality noise often leads to collapse and degraded generations. The resulting reward trajectories fluctuate widely, indicating that the model is frequently pushed toward regions of the noise space that are only weakly aligned with the target objective. In contrast, greedy selection stabilizes training by systematically steering updates toward high-reward regions, effectively implementing a simple yet powerful curriculum over the sampled noises. This targeted focus on promising directions helps maintain generation quality throughout training and prevents the large performance drops observed under random selection.

Sensitivity Analysis

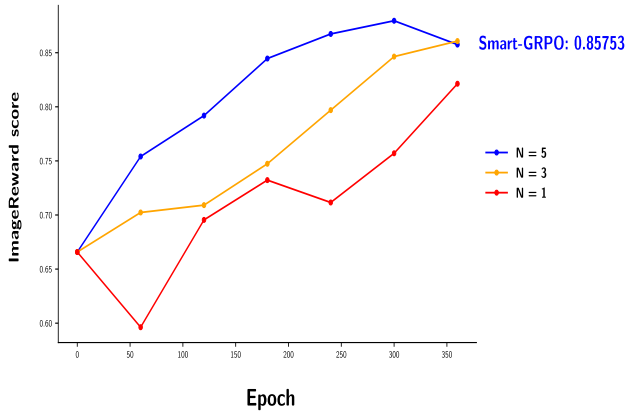

Figure 4: Sensitivity analysis for number of iterations used for Smart-GRPO. For 1 iteration, performance is unstable and fluctuates. When number of iterations increases, performance increases as iterations improve parameters more reliably.

We conducted a sensitivity analysis on the number of iterations to investigate how the choice of iterations influence training stability and reward performance.

When only a single iteration ($N = 1$) is used, Smart-GRPO effectively reduces to a one-shot update of the noise distribution. In this setting, the method provides small improvements over random perturbations, but the refinement process is too shallow: reward curves fluctuate noticeably across training, and final performance remains inconsistent.

Increasing to three iterations ($N = 3$) produces a marked change. Training becomes significantly more stable, with reward trajectories that are smoother and less noisy, and the models consistently achieve higher ImageReward and Aesthetic scores compared to $N = 1$. This suggests that multiple rounds of refinement allow the noise distribution to more reliably concentrate probability mass in promising regions.

With five iterations ($N = 5$), Smart-GRPO reaches its strongest performance. Both ImageReward and Aesthetic scores converge to their highest values, and optimization proceeds in a stable and predictable manner. Here, the repeated refinement cycles appear to provide the algorithm with enough opportunities to progressively adjust the distribution toward high-reward samples without overfitting to noise.

Taken together, these results highlight the importance of iterative refinement. By repeatedly resampling and updating, Smart-GRPO progressively guides the noise distribution toward high-quality solutions, improving both convergence speed and stability. While increasing N beyond 5 may yield further gains, it also comes with higher computational cost. Our experiments suggest that $N \in \{3, 5\}$ strikes a practical balance between efficiency and performance. Due to computational constraints, we did not explore higher values of N, leaving a more extensive exploration of this trade-off to future work.

Limitations

While Smart-GRPO demonstrates promising improvements, it also has several limitations. First, the effectiveness of our approach depends heavily on the choice of reward function. Many reward models are not well calibrated to evaluate poor-quality or highly noisy images, which constrains their ability to guide the noise selection process. For example, our experiments with PickScore (Kirstain et al. 2023) and CLIP-Score (Hessel et al. 2021) did not yield statistically significant gains, suggesting that these metrics may be ill-suited for reinforcement learning in high-noise regimes. A further lim-

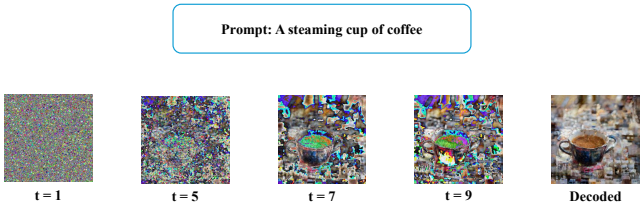

Figure 5: Intermediate approximations from Equation 6 during flow-matching generation of the prompt 'A steaming cup of coffee'. Starting from a noise level of 0.6 and decoded over 10 steps, earlier timesteps yield outputs resembling noise, while later timesteps progressively form low-quality images.

itation arises from the greedy approximation used in equation 6: as illustrated in Figure 5, this approximation does not hold well at earlier timesteps. Two issues follow: (1) the reward model is not designed to reliably score low-quality images, and (2) the early approximations themselves often fail to capture a meaningful representation of the final image. Exploring alternative metrics to evaluate high-quality noise could potentially be beneficial in improving model generations.

Second, due to computational constraints, we were unable to fully explore larger-scale experiments, longer training schedules, or higher values of K and N, which could further clarify the method's benefits.

Conclusion

We introduced Smart-GRPO, a simple yet effective framework for fine-tuning flow-matching generative models with reinforcement learning. By guiding noise sampling toward higher-reward regions, Smart-GRPO reduces wasted updates and improves both stability and convergence compared to existing methods. Importantly, this is the first approach to explicitly optimize the noise process for reinforcement learning in flow-matching models, offering a novel perspective on noise-aware training. We hope this work paves the way for future research on noise optimization, reward design, and scalable reinforcement learning for generative modeling.

Future Works

For future work, we envision several directions. One is to design or identify reward functions that are more robust at distinguishing subtle improvements in image quality, especially in the presence of noise. Another is to investigate alternative strategies for noise selection beyond mean and variance updates, such as adaptive sampling or learned proposal distributions. Finally, scaling experiments to larger models and more diverse benchmarks would provide a clearer picture of the generality and practical impact of Smart-GRPO.

Impact Statement

Smart-GRPO introduces a lightweight and efficient framework for reinforcement learning in flow-matching generative models. By directly optimizing the noise distribution, it reduces wasted training signal and achieves higher reward performance with fewer iterations. Because the method requires no architectural modifications and only a simple noise-selection loop, it can be seamlessly integrated into existing RLHF pipelines and deployed with modest computational resources.

In addition to these practical advantages, Smart-GRPO represents the first attempt to explicitly optimize the noise process for flow-matching models in the reinforcement learning setting. By reframing noise as an optimization variable, the method provides a novel perspective for improving generative modeling with reinforcement learning. This contribution opens a new line of research in noise-aware optimization, complementing advances in reward design and training objectives. Moreover, the generality of the approach suggests that similar techniques may be extended to diffusion models and other generative frameworks, paving the way for broader methodological innovations.

References

Batifol, S.; Blattmann, A.; Boesel, F.; Consul, S.; Diagne, C.; Dockhorn, T.; English, J.; English, Z.; Esser, P.; Kulal, S.; et al. 2025. FLUX. 1 Kontext: Flow Matching for In-Context Image Generation and Editing in Latent Space. *arXiv e-prints*, arXiv–2506.

Black, K.; Janner, M.; Du, Y.; Kostrikov, I.; and Levine, S. 2023. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*.

Esser, P.; Kulal, S.; Blattmann, A.; Entezari, R.; Müller, J.; Saini, H.; Levi, Y.; Lorenz, D.; Sauer, A.; Boesel, F.; et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.

Ghosh, D.; Hajishirzi, H.; and Schmidt, L. 2023. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152.

He, X.; Fu, S.; Zhao, Y.; Li, W.; Yang, J.; Yin, D.; Rao, F.; and Zhang, B. 2025. Tempflow-grpo: When timing matters for grpo in flow models. *arXiv preprint arXiv:2508.04324*.

Hessel, J.; Holtzman, A.; Forbes, M.; Bras, R. L.; and Choi, Y. 2021. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Kirstain, Y.; Polyak, A.; Singer, U.; Matiana, S.; Penna, J.; and Levy, O. 2023. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36: 36652–36663.

Li, J.; Cui, Y.; Huang, T.; Ma, Y.; Fan, C.; Yang, M.; and Zhong, Z. 2025a. Mixgrpo: Unlocking flow-based grpo efficiency with mixed ode-sde. *arXiv preprint arXiv:2507.21802*.

Li, Y.; Wang, Y.; Zhu, Y.; Zhao, Z.; Lu, M.; She, Q.; and Zhang, S. 2025b. BranchGRPO: Stable and Efficient GRPO

with Structured Branching in Diffusion Models. *arXiv preprint arXiv:2509.06040*.

Lipman, Y.; Chen, R. T. Q.; Ben-Hamu, H.; Nickel, M.; and Le, M. 2023. Flow Matching for Generative Modeling. *The Eleventh International Conference on Learning Representations*.

Liu, J.; Liu, G.; Liang, J.; Li, Y.; Liu, J.; Wang, X.; Wan, P.; Zhang, D.; and Ouyang, W. 2025. Flow-grpo: Training flow matching models via online rl. *arXiv preprint arXiv:2505.05470*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36: 53728–53741.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Wallace, B.; Dang, M.; Rafailov, R.; Zhou, L.; Lou, A.; Purushwalkam, S.; Ermon, S.; Xiong, C.; Joty, S.; and Naik, N. 2024. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8228–8238.

Wang, Y.; Li, Z.; Zang, Y.; Zhou, Y.; Bu, J.; Wang, C.; Lu, Q.; Jin, C.; and Wang, J. 2025. Pref-GRPO: Pairwise Preference Reward-based GRPO for Stable Text-to-Image Reinforcement Learning. *arXiv preprint arXiv:2508.20751*.

Xue, Z.; Wu, J.; Gao, Y.; Kong, F.; Zhu, L.; Chen, M.; Liu, Z.; Liu, W.; Guo, Q.; Huang, W.; et al. 2025. DanceGRPO: Unleashing GRPO on Visual Generation. *arXiv preprint arXiv:2505.07818*.

Yang, K.; Tao, J.; Lyu, J.; Ge, C.; Chen, J.; Shen, W.; Zhu, X.; and Li, X. 2024. Using human feedback to fine-tune diffusion models without any reward model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8941–8951.

Appendix

Hyperparameters

The hyperparameters used for training our models were largely copied from the hyperparameters used for Flow-GRPO. We only used 1 H100 GPU for training our model with GRPO. We initialized from the StabilityAI Stable Diffusion 3.5 Medium checkpoint.

All images were generated at a resolution of 512×512. During training, we used 10 sampling steps, while evaluation employed 40 sampling steps. We applied a classifier-free guidance scale of 4.5.

Each training batch contained 4 images per prompt, with an effective training batch size of 4. To ensure balanced gradient updates, we set the number of batches per epoch to 8, which yielded an even number of batches per epoch. Gradient accumulation was configured such that two updates occurred per epoch. The test batch size was fixed to 16.

We trained with 1 inner epoch per outer epoch, and sampled timesteps with a fraction of 0.99. Optimization included a KL loss term weighted by $\beta = 0.04$. We enabled exponential moving average (EMA) of model weights.

We saved model checkpoints every 60 epochs and performed evaluation at the same frequency.