A. Additional Implementation Details

Our implementation adheres closely to the baseline methods, with deviations limited to the key techniques described in Sec. 4.4 and Sec. 4.5.

A.1. FLUX.1-dev Experiments

Main experiments. Our setup follows MixGRPO [19], training and evaluating on prompts drawn from the HPDv2 [42] dataset. For the reward functions, we utilize an ensemble of HPSv2.1 [42], PickScore [15], ImageReward [43], and UnifiedReward [41]. PickScore is normalized in the same way as in MixGRPO [19]. Multi-reward advantages are computed by averaging the mean of the individual reward advantages.

We optimize the model using AdamW [25] with a learning rate of 1×10^{-5} and a weight decay of 1×10^{-4}. Training proceeds for 300 iterations, each comprising 4 gradient steps with a global batch size of 8 per step and a group size of 12. We do not maintain an exponential moving average (EMA) of weights during training.

We set the number of timestep-noise pairs to $N_{\text{MC}} = 4$. For training with 25 sampling steps during rollout, importance ratios are clipped to 6×10^{-3}. For training with 16 sampling steps, importance ratios are clipped to 1×10^{-2} and advantages are soft-clipped to 2. No KL penalty is applied in either configuration.

During training rollout, we use a resolution of 720×720. Main experiments are conducted with both 16 and 25 sampling steps. At evaluation time, we scale this to 50 steps at a 1024×1024 resolution. To mitigate reward hacking while preserving strong generation quality during evaluation, we adopt the MixGRPO [19] hybrid sampling strategy. Specifically, the trained model is used for the first $p_{\text{mix}}T$ steps (with $p_{\text{mix}} = 0.8$), and the original base model completes the remainder.

Ablation studies. We only use 25 sampling steps in ablation studies. All other configurations are consistent with those used in the main experiments.

A.2. SD 3.5 M Experiments

Main experiments. We adopt a multi-stage training curriculum from DiffusionNFT [47], which leverages diverse reward functions and prompt datasets. For multi-reward advantage estimation, we first aggregate individual rewards via averaging, and then compute advantages using these aggregated values.

We optimize the model using LoRA [12] ($r = 32, \alpha = 64$) and the AdamW optimizer with a learning rate of 3×10^{-4} and a weight decay of 1×10^{-4}. Across all stages, training is conducted with a global batch size of 48 per gradient step and a group size of 24, matching the per-step configuration of DiffusionNFT [47].

To maintain consistency with DiffusionNFT, we favor fully online training with advantage soft-clipping, falling back to importance ratio clipping or a KL penalty when this alone is insufficient to achieve optimal performance. Unlike DiffusionNFT, we avoid reusing optimizer states between stages and do not maintain an EMA of weights.

Our multi-stage training curriculum is detailed below:

- **Stages 1 and 3.** The model is trained on Pick-a-Pic [15] using an ensemble of HPSv2.1 [42], PickScore [15], and CLIPScore [9]. PickScore is normalized in the same way as in DiffusionNFT [47]. In line with the baselines, each iteration performs only 1 gradient step. In both stages, we run 150 iterations (150 gradient steps), use $N_{\text{MC}} = 4$ timestep–noise pairs, and soft-clip advantages to 3. No importance ratio clipping or KL penalty is applied.
- **Stages 2 and 4.** We add GenEval [6] to the three initial rewards. To improve performance through importance ratio clipping, each iteration performs 2 gradient steps. In Stage 2, training runs for 100 iterations (200 gradient steps) on Flow-GRPO [23]'s prompt sets with importance ratios clipped to 2×10^{-3}. In Stage 4, training runs for 25 iterations (50 gradient steps) with importance ratios clipped to 4×10^{-3}. In both stages, we use $N_{\text{MC}} = 10$. No KL penalty or advantage soft-clipping is applied.
- **Stage 5.** We add OCR [2] to the three initial rewards. To preserve capabilities acquired during prior training via KL penalty, each iteration performs 2 gradient steps. Training runs for 15 iterations (30 gradient steps) on Flow-GRPO [23]'s prompt sets with $N_{\text{MC}} = 10$ and a KL coefficient of 0.3. No importance ratio clipping or advantage soft-clipping is applied.

During both training rollout and evaluation time, we use 40 sampling steps at a resolution of 512×512. Beyond the reward functions used during training, we further evaluate the trained model using out-of-domain metrics, including CLIPScore [9], UnifiedReward [41], and Aesthetics [32].

To demonstrate the superior efficiency of our method, we compare its gradient step counts and NFE against those of DiffusionNFT in Tab. 5.

Table 5. **Comparison of gradient step counts and NFEs across training stages.** Our method delivers a $3\times$ speedup over DiffusionNFT in gradient steps, while also requiring fewer function evaluations (NFE) per step on average. DiffusionNFT reports per-stage step counts as approximate values due to its aggressive use of early stopping, whereas our counts are exact.

Stage	DiffusionNFT		V-GRPO	
	#Steps	NFE	#Steps	NFE
1 (human preferences)	800	120	150	48
2 (GenEval)	300	120	200	60
3 (human preferences)	200	120	150	48
4 (GenEval)	200	120	50	60
5 (OCR)	100	120	30	60
Total	1700	120	580	53.8

Table 6. **Evaluation results for single-reward experiments on SD 3.5 M.** All methods disable CFG during both training and evaluation. For models trained with the OCR reward, CFG is re-enabled when evaluating non-OCR rewards, following DiffusionNFT. Baseline results are sourced from DiffusionNFT. Gray-colored: In-domain reward. **Bold**: best; Underline: second best.

Method	#Steps	Rule-Based		Model-Based					
		GenEval	OCR	PickScore	CLIPScore	HPSv2.1	Aesthetic	ImgRwd	UniRwd
SD 3.5 M (w/o CFG)	—	0.24	0.12	0.2051	0.237	0.204	5.13	-0.58	2.02
+ CFG	—	0.63	0.59	0.2234	0.285	0.279	5.36	0.85	3.03
+ FlowGRPO	4K	0.97	0.30	0.2178	0.277	0.248	5.15	**0.74**	2.87
+ DiffusionNFT	1K	**0.98**	0.36	**0.2192**	0.271	**0.251**	**5.33**	0.68	2.91
+ V-GRPO	500	0.97	**0.39**	0.2155	**0.283**	0.228	5.12	0.42	**3.19**
+ FlowGRPO	1K	0.66	0.96	**0.2194**	0.280	**0.257**	5.18	0.31	2.86
+ DiffusionNFT	150	0.54	0.97	0.2163	**0.281**	0.246	5.19	**0.37**	2.81
+ V-GRPO	25	0.47	**0.98**	0.2170	0.277	0.243	**5.21**	0.28	**2.98**
+ FlowGRPO	4K	0.54	0.60	0.2362	0.257	0.295	**6.42**	1.17	3.17
+ DiffusionNFT	2K	0.53	**0.64**	**0.2403**	**0.270**	**0.315**	6.17	1.29	3.40
+ V-GRPO	300	**0.66**	0.62	**0.2403**	0.267	0.308	**6.42**	**1.30**	**3.47**

Single-reward experiments. In our GenEval single-reward experiments, hyperparameters follow those in the Stage 2 of the main experiments, with training running for 250 iterations (500 gradient steps). For OCR, training runs for 25 iterations (25 gradient steps), using $N_{\text{MC}} = 10$ with advantages soft-clipped to 4. For PickScore, training runs for 300 iterations (300 gradient steps), using $N_{\text{MC}} = 4$ with advantages soft-clipped to 3. Both OCR and PickScore experiments perform 1 gradient step per iteration. All other configurations are consistent with those used in the main experiments.

Ablation studies. Unless otherwise stated, all implementation details are the same as the main experiments.

B. Additional Results

Additional qualitative examples from the FLUX.1-dev [16] and SD 3.5 M [3] main experiments are illustrated in Fig. 7 and Fig. 8, respectively.

Quantitative comparisons of single-reward experiments on SD 3.5 M are reported in Tab. 6.

In Fig. 5, we ablate the proposed surrogate variance reduction techniques on SD 3.5 M. While these techniques are collectively beneficial, no single component is individually critical. In Fig. 6, we examine the effect of prediction parameterization on the adaptive loss weighting technique. $\boldsymbol{\epsilon}$-prediction leads to severe training collapse, whereas $\mathbf{v}$-prediction remains stable but yields slightly slower convergence than $\mathbf{x}$-prediction.

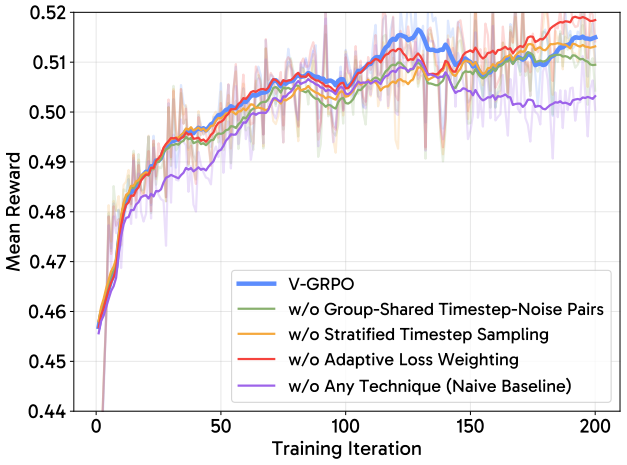

Figure 5. **Ablation studies of surrogate variance reduction techniques.** Implementation details follow those of Stage-1 training in the SD 3.5 main experiments.

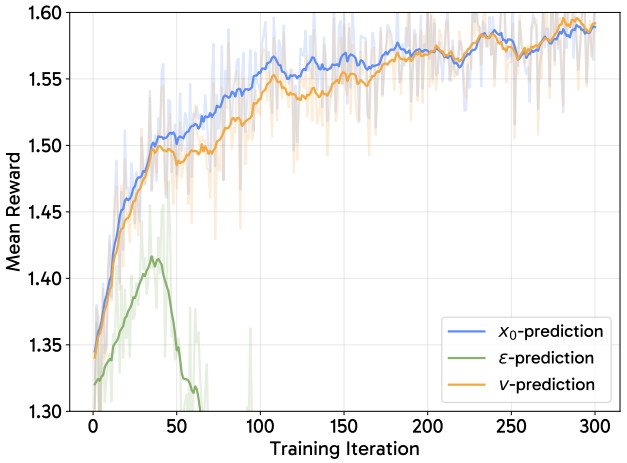

Figure 6. **Ablation studies of alternative reparameterizations of model predictions.** Implementation details follow those used in the FLUX.1-dev experiments.

Figure 7. **Qualitative comparison from the FLUX.1-dev main experiments.** V-GRPO achieves superior performance in alignment, coherence, and style. In the fourth example, it demonstrates strong world knowledge.

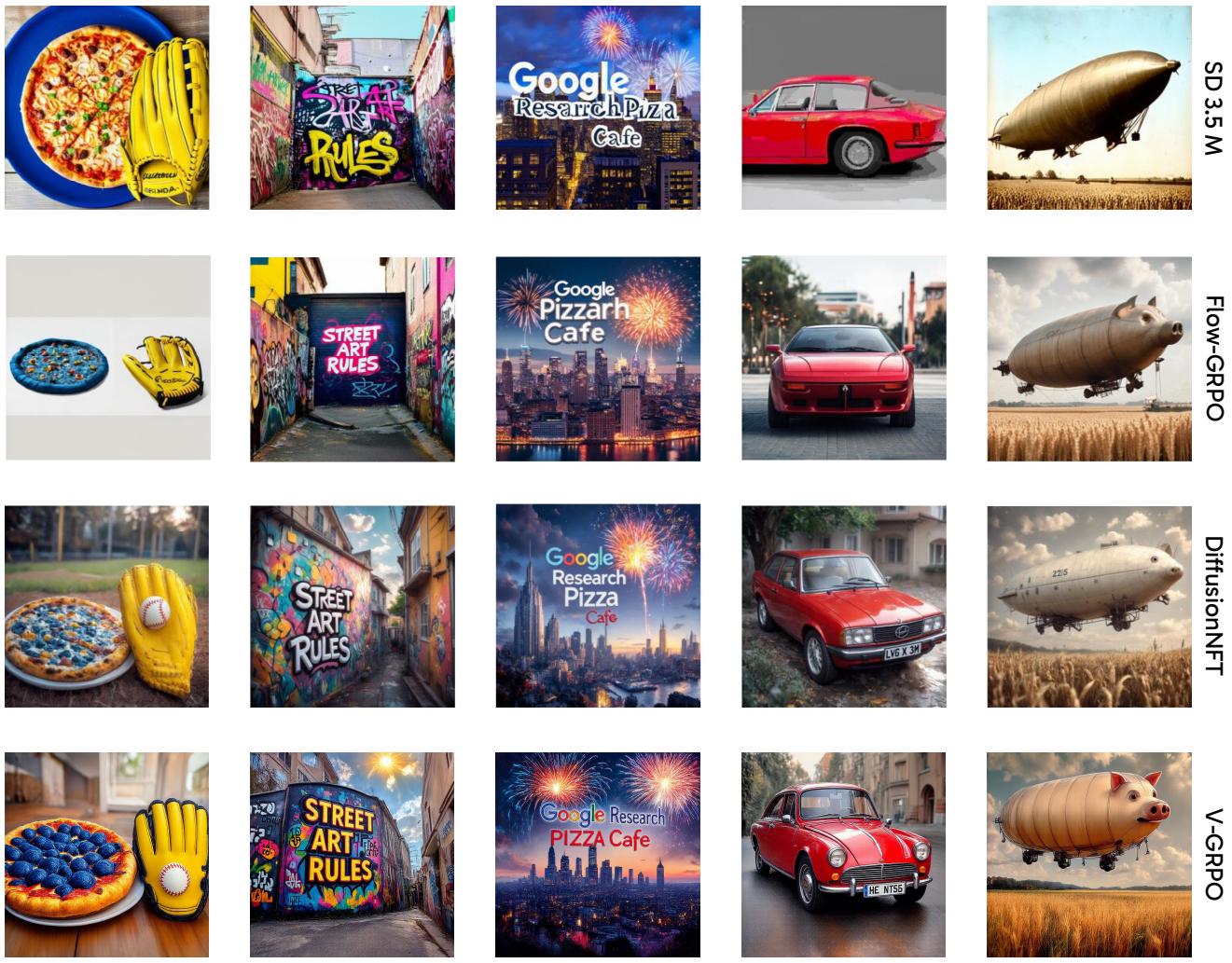

Figure 8. **Qualitative comparison from the SD 3.5 M main experiments.** V-GRPO achieves superior performance in alignment, coherence, and style.