



DanceGRPO: Unleashing GRPO on Visual Generation

Zeyue Xue^{1,2}, Jie Wu^{1†}, Yu Gao¹, Fangyuan Kong¹, Lingting Zhu², Mengzhao Chen²,
Zhiheng Liu², Wei Liu¹, Qiushan Guo¹, Weilin Huang^{1†}, Ping Luo^{2†}

¹ByteDance Seed, ²The University of Hong Kong

†Corresponding authors, †Project lead

Abstract

Recent breakthroughs in generative models—particularly diffusion models and rectified flows—have revolutionized visual content creation, yet aligning model outputs with human preferences remains a critical challenge. Existing reinforcement learning (RL)-based methods for visual generation face critical limitations: incompatibility with modern Ordinary Differential Equations (ODEs)-based sampling paradigms, instability in large-scale training, and lack of validation for video generation. This paper introduces **DanceGRPO**, the first unified framework to adapt Group Relative Policy Optimization (GRPO) to visual generation paradigms, unleashing *one* unified RL algorithm across *two* generative paradigms (diffusion models and rectified flows), *three* tasks (text-to-image, text-to-video, image-to-video), *four* foundation models (Stable Diffusion, HunyuanVideo, FLUX, SkyReels-I2V), and *five* reward models (image/video aesthetics, text-image alignment, video motion quality, and binary reward). To our knowledge, DanceGRPO is the **first** RL-based unified framework capable of seamless adaptation across diverse generative paradigms, tasks, foundational models, and reward models. DanceGRPO demonstrates consistent and substantial improvements, which outperform baselines by up to 181% on benchmarks such as HPS-v2.1, CLIP Score, VideoAlign, and GenEval. Notably, DanceGRPO not only can stabilize policy optimization for complex video generation, but also enables generative policy to better capture denoising trajectories for Best-of-N inference scaling and learn from sparse binary feedback. Our results establish DanceGRPO as a robust and versatile solution for scaling Reinforcement Learning from Human Feedback (RLHF) tasks in visual generation, offering new insights into harmonizing reinforcement learning and visual synthesis. The code will be released.

Date: May 1, 2025

Project Page: <https://dancegrpo.github.io/>

1 Introduction

Recent advances in generative models—particularly diffusion models [1–4] and rectified flows [5–7]—have transformed visual content creation by improving output quality and versatility in image and video generation. While pretraining establishes foundational data distributions, integrating human feedback during training proves critical for aligning outputs with human preferences and aesthetic criteria [8]. Existing methods face notable limitations: ReFL [9] relies on differentiable reward models, which introduce VRAM inefficiency in video generation and require several extensive engineering efforts, while DPO variants (Diffusion-DPO [10], Flow-DPO [11]) achieve only marginal visual quality improvements. Reinforcement learning (RL)-based methods [12, 13], which optimize rewards as black-box objectives, offer potential solutions but introduce three

unresolved challenges: (1) the Ordinary Differential Equations (ODEs)-based sampling of rectified flow models conflict with Markov Decision Process formulations; (2) prior policy gradient approaches (DDPO [14], DPOK [15]) show instability when scaling beyond small datasets (e.g., <100 prompts); and (3) existing methods remain unvalidated for video generation tasks.

This work addresses these gaps by reformulating the sampling of diffusion models and rectified flows via Stochastic Differential Equations (SDEs) and applying Group Relative Policy Optimization (GRPO) [16, 17] to stabilize the training process. GRPO is a recent breakthrough that enhances the performance of large language models (LLMs) on complex tasks such as mathematics and coding. In this paper, we pioneer the adaptation of GRPO to visual generation tasks through the **DanceGRPO** framework, establishing a "harmonious dance" between GRPO and visual generation tasks.

We present a comprehensive study of **one** algorithm, DanceGRPO across **two** generative paradigms (diffusion/rectified flows), **three** tasks (text-to-image, text-to-video, image-to-video), **four** foundation models (Stable Diffusion [2], HunyuanVideo [18], FLUX [19], SkyReels-I2V [20]), and **five** reward models (image/video aesthetics, text-image alignment, video motion quality, and binary thresholds).

Our contributions can be summarized as follows:

- **Unified and Pioneering.** To the best of our knowledge, we are the first to extend GRPO to diffusion models and rectified flows, accomplishing multiple visual generation tasks within a unified RL framework. We achieve seamless integration between GRPO and visual generation tasks by carefully reformulating the SDEs, selecting appropriate optimized timesteps, and initializing noise and noise scales.
- **Generalization and Scalability.** To our knowledge, DanceGRPO is the first RL-based unified framework capable of seamless adaptation across diverse generative paradigms, tasks, foundational models, and reward models. Unlike prior RL algorithms, primarily validated on text-to-image diffusion models on small-scale datasets, DanceGRPO demonstrates robust performance on large-scale datasets, showcasing both scalability and practical applicability.
- **High Effectiveness.** Our experiments demonstrate that DanceGRPO achieves significant performance gains, outperforming baselines by up to 181% across multiple academic benchmarks, including HPS-v2.1 [21], CLIP score [22], VideoAlign [11], and GenEval [23], in visual generation tasks. Notably, DanceGRPO also enables models to learn the denoising trajectory in Best-of-N inference scaling. We also make some initial attempts to enable models to capture the distribution of binary (0/1) reward models, showing its ability to capture sparse, thresholding feedback.

2 Approach

2.1 Preliminary

Diffusion Model [1]. A diffusion process gradually destroys an observed datapoint \mathbf{x} over timestep t , by mixing data with noise, and the forward process of the diffusion model can be defined as :

$$\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}), \quad (1)$$

and α_t and σ_t denote the noise schedule. The noise schedule is designed in a way such that \mathbf{z}_0 is close to clean data and \mathbf{z}_1 is close to Gaussian noise. To generate a new sample, we initialize the sample \mathbf{z}_1 and define the sample equation of the diffusion model given the denoising model output $\hat{\boldsymbol{\epsilon}}$ at time step t :

$$\mathbf{z}_s = \alpha_s \hat{\mathbf{x}} + \sigma_s \hat{\boldsymbol{\epsilon}}, \quad (2)$$

where $\hat{\mathbf{x}}$ can be derived via Eq.(1) and then we can reach a lower noise level s . This is also a DDIM sampler [24].

Rectified Flow [6]. In rectified flow, we view the forward process as a linear interpolation between the data \mathbf{x} and a noise term $\boldsymbol{\epsilon}$:

$$\mathbf{z}_t = (1 - t)\mathbf{x} + t\boldsymbol{\epsilon}, \quad (3)$$

where ϵ is always defined as a Gaussian noise. We define the $\mathbf{u} = \epsilon - \mathbf{x}$ as the "velocity" or "vector field". Similar to diffusion model, given the denoising model output $\hat{\mathbf{u}}$ at time step t , we can reach a lower noise level s by:

$$\mathbf{z}_s = \mathbf{z}_t + \hat{\mathbf{u}} \cdot (s - t). \quad (4)$$

Analysis. Although the diffusion model and rectified flow have different theoretical foundations, in practice, they are two sides of a coin, as shown in the following formula:

$$\tilde{\mathbf{z}}_s = \tilde{\mathbf{z}}_t + \text{Network output} \cdot (\eta_s - \eta_t). \quad (5)$$

For ϵ -prediction (a.k.a. diffusion model), from Eq.(2), we have $\tilde{\mathbf{z}}_s = \mathbf{z}_s/\alpha_s$, $\tilde{\mathbf{z}}_t = \mathbf{z}_t/\alpha_t$, $\eta_s = \sigma_s/\alpha_s$, and $\eta_t = \sigma_t/\alpha_t$. For rectified flows, we have $\tilde{\mathbf{z}}_s = \mathbf{z}_s$, $\tilde{\mathbf{z}}_t = \mathbf{z}_t$, $\eta_s = s$, and $\eta_t = t$ from Eq.(4).

2.2 DanceGRPO

In this section, we first formulate the sampling processes of diffusion models and rectified flows as Markov Decision Processes. Then, we introduce the sampling SDEs and the algorithm of DanceGRPO.

Denoising as a Markov Decision Process. Following DDPO [14], we formulate the denoising process of the diffusion model and rectified flow as a Markov Decision Process (MDP):

$$\begin{aligned} \mathbf{s}_t &\triangleq (\mathbf{c}, t, \mathbf{z}_t), \quad \pi(\mathbf{a}_t | \mathbf{s}_t) \triangleq p(\mathbf{z}_{t-1} | \mathbf{z}_t, \mathbf{c}), \quad P(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t) \triangleq (\delta_{\mathbf{c}}, \delta_{t-1}, \delta_{\mathbf{z}_{t-1}}) \\ \mathbf{a}_t &\triangleq \mathbf{z}_{t-1}, \quad R(\mathbf{s}_t, \mathbf{a}_t) \triangleq \begin{cases} r(\mathbf{z}_0, \mathbf{c}), & \text{if } t=0 \\ 0, & \text{otherwise} \end{cases}, \quad \rho_0(\mathbf{s}_0) \triangleq (p(\mathbf{c}), \delta_T, \mathcal{N}(\mathbf{0}, \mathbf{I})), \end{aligned} \quad (6)$$

where \mathbf{c} is the prompt, and $\pi(\mathbf{a}_t | \mathbf{s}_t)$ is the probability from z_t to z_{t-1} . And δ_y is the Dirac delta distribution with nonzero density only at y . Trajectories consist of T timesteps, after which P leads to a termination state. $r(\mathbf{z}_0, \mathbf{c})$ is the reward model, which is always parametrized by a Vision-Language model (such as CLIP [22] and Qwen-VL [25]).

Formulation of Sampling SDEs. Since GRPO requires stochastic exploration through multiple trajectory samples, where policy updates depend on the trajectory probability distribution and their associated reward signals, we unify the sampling processes of the diffusion model and rectified flows into the form of SDEs. For the diffusion model, as demonstrated in [26, 27], the forward SDE is given by: $d\mathbf{z}_t = f_t \mathbf{z}_t dt + g_t d\mathbf{w}$. The corresponding reverse SDE can be expressed as:

$$d\mathbf{z}_t = \left(f_t \mathbf{z}_t - \frac{1 + \varepsilon_t^2}{2} g_t^2 \nabla \log p_t(\mathbf{z}_t) \right) dt + \varepsilon_t g_t d\mathbf{w}, \quad (7)$$

where $d\mathbf{w}$ is a Brownian motion, and ε_t introduces the stochasticity during sampling.

Similarly, the forward ODE of rectified flow is: $d\mathbf{z}_t = \mathbf{u}_t dt$. The generative process reverses the ODE in time. However, this deterministic formulation cannot provide the stochastic exploration required for GRPO. Inspired by [28–30], we introduce an SDE case during the reverse process as follows:

$$d\mathbf{z}_t = (\mathbf{u}_t - \frac{1}{2} \varepsilon_t^2 \nabla \log p_t(\mathbf{z}_t)) dt + \varepsilon_t d\mathbf{w}, \quad (8)$$

where ε_t also introduces the stochasticity during sampling. Given a normal distribution $p_t(\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t | \alpha_t \mathbf{x}, \sigma_t^2 I)$, we have $\nabla \log p_t(\mathbf{z}_t) = -(\mathbf{z}_t - \alpha_t \mathbf{x})/\sigma_t^2$. We can insert this into the above two SDEs and obtain the $\pi(\mathbf{a}_t | \mathbf{s}_t)$. More theoretical analysis can be found in Appendix B.

Algorithm. Motivated by Deepseek-R1 [16], given a prompt \mathbf{c} , generative models will sample a group of outputs $\{\mathbf{o}_1, \mathbf{o}_2, \dots, \mathbf{o}_G\}$ from the model $\pi_{\theta_{old}}$, and optimize the policy model π_θ by maximizing the following objective function:

$$\mathcal{J}(\theta) = \mathbb{E}_{\substack{\{\mathbf{o}_i\}_{i=1}^G \sim \pi_{\theta_{old}}(\cdot | \mathbf{c}) \\ \mathbf{a}_{t,i} \sim \pi_{\theta_{old}}(\cdot | \mathbf{s}_{t,i})}} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{T} \sum_{t=1}^T \min \left(\rho_{t,i} A_i, \text{clip}(\rho_{t,i}, 1 - \epsilon, 1 + \epsilon) A_i \right) \right], \quad (9)$$

where $\rho_{t,i} = \frac{\pi_\theta(\mathbf{a}_{t,i}|\mathbf{s}_{t,i})}{\pi_{\theta_{old}}(\mathbf{a}_{t,i}|\mathbf{s}_{t,i})}$, and $\pi_\theta(\mathbf{a}_{t,i}|\mathbf{s}_{t,i})$ is the policy function is MDP for output \mathbf{o}_i at time step t , ϵ is a hyper-parameter, and A_i is the advantage function, computed using a group of rewards $\{r_1, r_2, \dots, r_G\}$ corresponding to the outputs within each group:

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \dots, r_G\})}{\text{std}(\{r_1, r_2, \dots, r_G\})}. \quad (10)$$

Due to reward sparsity in practice, we apply the same reward signal across all timesteps during optimization. While traditional GRPO formulations employ KL-regularization to prevent reward over-optimization, we empirically observe minimal performance differences when omitting this component. So, we omit the KL-regularization item by default. The full algorithm can be found in Algorithm 1. We also introduce how to train with Classifier-Free Guidance (CFG) [31] in Appendix C. In summary, we formulate the sampling processes of diffusion model and rectified flow as MDPs, use SDE sampling equations, adopt a GRPO-style objective, and generalize to text-to-image, text-to-video, and image-to-video generation tasks.

Initialization Noise. In the DanceGRPO framework, the initialization noise constitutes a critical component. Previous RL-based approaches like DDPO use different noise vectors to initialize training samples. However, as shown in Figure 8 in Appendix F, assigning different noise vectors to samples with the same prompts always leads to reward hacking phenomena in video generation, including training instability. Therefore, in our framework, we assign shared initialization noise to samples originating from the same textual prompts.

Timestep Selection. While the denoising process can be rigorously formalized within the MDP framework, empirical observations reveal that subsets of timesteps within a denoising trajectory can be omitted without compromising performance. This reduction in computational steps enhances efficiency while maintaining output quality, as further analyzed in Section 3.6.

Incorporating Multiple Reward Models. In practice, we employ more than one reward model to ensure more stable training and higher-quality visual results. As illustrated in Figure 9 in Appendix, models trained exclusively with HPS-v2.1 rewards [21] tend to generate unnatural ("oily") outputs, whereas incorporating CLIP scores helps preserve more realistic image characteristics. Rather than directly combining rewards, we aggregate advantage functions, as different reward models often operate on different scales. This approach stabilizes optimization and leads to more balanced generations.

Extension on Best-of-N Inference Scaling. As outlined in Section 3.6, our methodology prioritizes the use of efficient samples—specifically, those associated with the top k and bottom k candidates selected by the Best-of- N sampling. This selective sampling strategy optimizes training efficacy by focusing on high-reward and critical low-reward regions of the solution space. We use brute-force search to generate these samples. While alternative approaches, such as tree search or greedy search, remain promising avenues for further exploration, we defer their systematic integration to future research.

2.3 Application to Different Tasks with Different Rewards

We verify the effectiveness of our algorithm in two generative paradigms (diffusion/rectified flow) and three tasks (text-to-image, text-to-video, image-to-video). For this, we choose four fundamental models (Stable Diffusion [2], HunyuanVideo [18], FLUX [19], SkyReels-I2V [20]) for the experiment. All of these methods can be precisely constructed within the framework of MDP during their sampling process. This allows us to unify the theoretical bases across these tasks and improve them via DanceGRPO. To our knowledge, this is the first work to apply the unified framework to diverse visual generation tasks.

We use five reward models to optimize visual generation quality: (1) Image Aesthetics quantifies visual appeal using a pretrained model fine-tuned on human-rated data [21]; (2) Text-image Alignment employs CLIP [22] to maximize cross-modal consistency between prompts and outputs; (3) Video Aesthetics Quality extends image evaluation to temporal domains using VLMs [11, 25], assessing frame quality and coherence; (4) Video Motion Quality evaluates motion realism through physics-aware VLM [11] analysis of trajectories and deformations; (5) Thresholding Binary Reward employs a binary mechanism motivated by [16], where rewards are discretized via a fixed threshold (values exceeding the threshold receive 1, others 0), specifically designed to evaluate generative models' ability to learn abrupt reward distributions under threshold-based optimization.

Algorithm 1 DanceGRPO Training Algorithm

Require: Initial policy model π_θ ; reward models $\{R_k\}_{k=1}^K$; prompt dataset \mathcal{D} ; timestep selection ratio τ ; total sampling steps T

Ensure: Optimized policy model π_θ

- 1: **for** training iteration = 1 **to** M **do**
- 2: Sample batch $\mathcal{D}_b \sim \mathcal{D}$ ▷ Batch of prompts
- 3: Update old policy: $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
- 4: **for** each prompt $\mathbf{c} \in \mathcal{D}_b$ **do**
- 5: Generate G samples: $\{\mathbf{o}_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{c})$ with the same random initialization noise
- 6: Compute rewards $\{r_i^k\}_{i=1}^G$ using each R_k
- 7: **for** each sample $i \in 1..G$ **do**
- 8: Calculate multi-reward advantage: $A_i \leftarrow \sum_{k=1}^K \frac{r_i^k - \mu^k}{\sigma^k}$ ▷ μ^k, σ^k per-reward statistics
- 9: **end for**
- 10: Subsample $\lceil \tau T \rceil$ timesteps $\mathcal{T}_{\text{sub}} \subset \{1..T\}$
- 11: **for** $t \in \mathcal{T}_{\text{sub}}$ **do**
- 12: Update policy via gradient ascent: $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{J}$
- 13: **end for**
- 14: **end for**
- 15: **end for**

Table 1 Comparison of alignment methods across key capabilities. VideoGen: Video generation generalization. Scalability: Scalability to datasets with a large number of prompts. Reward \uparrow indicates a significant reward improvement. RFs: Applicable to Rectified Flows. No Diff-Reward: Don’t need differentiable reward models.

Method	RL-based	VideoGen	Scalability	Reward \uparrow	RFs	No Diff-Reward
DDPO/DPOK	✓	✗	✗	✓	✗	✓
ReFL	✗	✗	✓	✓	✓	✗
DPO	✗	✓	✓	✗	✓	✓
Ours	✓	✓	✓	✓	✓	✓

2.4 Comparisons with DDPO, DPOK, ReFL, and DPO

As evidenced by our comprehensive capability matrix in Table 1, DanceGRPO establishes new standards for diffusion model alignment. Our method achieves full-spectrum superiority across all evaluation dimensions: (1) seamless video generation, (2) large-scale dataset scalability, (3) significant reward improvements, (4) native compatibility with rectified flows, and (5) independence from differentiable rewards. This integrated capability profile – unobtainable by any single baseline method (DDPO/DPOK/ReFL/DPO) – enables simultaneous optimization across multiple generative domains while maintaining training stability. More comparisons can be found in Appendix D.

3 Experiments

3.1 General Setup

Text-to-Image Generation. We employ Stable Diffusion v1.4, FLUX, and HunyuanVideo-T2I (using one latent frame in HunyuanVideo) as foundation models, with HPS-v2.1 [21] and CLIP score [22]—alongside their binary rewards—serving as reward models. A curated prompt dataset, balancing diversity and complexity, guides optimization. For evaluation, we select 1,000 test prompts to assess CLIP scores and Pick-a-Pic performance in Section 3.2. We use the official prompts for GenEval and HPS-v2.1 benchmark.

Text-to-Video Generation. Our foundation model is HunyuanVideo [18], with reward signals derived from VideoAlign [11]. Prompts are curated using the VidProM [32] dataset, and an additional 1,000 test prompts are filtered to evaluate the VideoAlign scores in Section 3.3.

Image-to-Video Generation. We use SkyReels-I2V [20] as our foundation model. VideoAlign [11] serves as the primary reward metric, while the prompt dataset, constructed via VidProM [32], is paired with reference images synthesized by HunyuanVideo-T2I [18] to ensure conditional fidelity. An additional 1000 test prompts

are filtered to evaluate the VideoAlign score in Section 3.4.

Experimental Settings. We implemented all models with scaled computational resources appropriate to task complexity: 32 H800 GPUs for flow-based text-to-image models, 8 H800 GPUs for Stable Diffusion variants, 64 H800 GPUs for text-to-video generation systems, and 32 H800 GPUs for image-to-video transformation architectures. We develop our framework based on FastVideo [33, 34]. Comprehensive hyperparameter configurations and training protocols are detailed in Appendix A. We always use more than 10,000 prompts to optimize the models. All reward curves presented in our paper are plotted using a moving average for smoother visualization. We use ODEs-based samplers for evaluation and visualization.

Table 2 Results on Stable Diffusion v1.4. This table presents the performance of three Stable Diffusion variants: (1) the base model, (2) the model trained with HPS score, and (3) the model optimized with both HPS and CLIP scores. For evaluation, we report HPS-v2.1 and GenEval scores using their official prompts, while CLIP score and Pick-a-Pic metrics are computed on our test set of 1,000 prompts.

Models	HPS-v2.1 [21]	CLIP Score [22]	Pick-a-Pic [35]	GenEval [23]
Stable Diffusion	0.239	0.363	0.202	0.421
Stable Diffusion with HPS-v2.1	0.365	0.380	0.217	0.521
Stable Diffusion with HPS-v2.1&CLIP Score	0.335	0.395	0.215	0.522

Table 3 Results on FLUX. In this table, we show the results of FLUX, FLUX trained with HPS score, and FLUX trained with both HPS score and CLIP score. We use the same evaluation prompts as Table 2.

Models	HPS-v2.1 [21]	CLIP Score [22]	Pick-a-Pic [35]	GenEval [23]
FLUX	0.304	0.405	0.224	0.659
FLUX with HPS-v2.1	0.372	0.376	0.230	0.561
FLUX with HPS-v2.1&CLIP Score	0.343	0.427	0.228	0.687

Table 4 Comparison of different methods trained on **Table 5** The results of HunyuanVideo on Videoalign and diffusion and solely (not combined) with HPS score and VisionReward trained with VideoAlign VQ&MQ. "Baseline" denotes the original results of line" denotes the original results of HunyuanVideo. We Stable Diffusion. More comparisons with DDPO can be use the weighted sum of the probability for VisionReward-found in Appendix E.

Approach	Baseline	Ours	DDPO	ReFL	DPO
HPS-v2.1	0.239	0.365	0.297	0.357	0.241
CLIP Score	0.363	0.421	0.381	0.418	0.367

Benchmarks	VQ	MQ	TA	VisionReward
Baseline	4.51	1.37	1.75	0.124
Ours	7.03 (+56%)	3.85 (+181%)	1.59	0.128

3.2 Text-to-Image Generation

Stable Diffusion. Stable Diffusion v1.4, a diffusion-based text-to-image generation framework, comprises three core components: a UNet architecture for iterative denoising, a CLIP-based text encoder for semantic conditioning, and a variational autoencoder (VAE) for latent space modeling. As demonstrated in Table 2 and Figure 1(a), our proposed method, DanceGRPO, achieves a significant improvement in reward metrics, elevating the HPS score from 0.239 to 0.365, as well as the CLIP Score from 0.363 to 0.395. We also take the metric like the Pick-a-Pic [35] and GenEval [23] to evaluate our method. The results confirm the effectiveness of our method. Moreover, as shown in Table 4, our method exhibits the best performance in terms of metrics compared to other methods. We implement DPO as an online version following [36].

Building on insights from rule-based reward models such as DeepSeek-R1, we conduct preliminary experiments with a binary reward formulation. By thresholding the continuous HPS reward at 0.28—assigning (CLIP score at 0.39) a value of 1 for rewards above this threshold and 0 otherwise, we construct a simplified reward model. Figure 3(a) illustrates that DanceGRPO effectively adapts to this discretized reward distribution, despite the inherent simplicity of the thresholding approach. These results indicate the effectiveness of binary reward models in visual generation tasks. In the future, we will also strive to explore more powerful rule-based visual reward models, for example, making judgments through a multimodal large language model.

Best-of-N Inference Scaling. We explore sample efficiency through Best-of-N inference scaling using Stable Diffusion, as detailed in Section 2.2. By training the model on subsets of 16 samples (with the top 8 and bottom 8 rewards) selected from progressively larger pools (16, 64, and 256 samples per prompt), we evaluate the impact of sample curation on convergence dynamics about Stable Diffusion. Figure 4(a) reveals that Best-of-N scaling substantially accelerates convergence. This underscores the utility of strategic sample selection in reducing training overhead while maintaining performance. For alternative approaches, such as tree search or greedy search, we defer their systematic integration to future research.

FLUX. FLUX.1-dev is a flow-based text-to-image generation model that advances the state-of-the-art across multiple benchmarks, leveraging a more complex architecture than Stable Diffusion. To optimize performance, we integrate two reward models: HPS score and CLIP score. As illustrated in Figure 1(b) and Table 3, the proposed training paradigm achieves significant improvements across all reward metrics.

HunyuanVideo-T2I. HunyuanVideo-T2I is a text-to-image adaptation of the HunyuanVideo framework, reconfigured by reducing the number of latent frames to one. This modification transforms the original video generation architecture into a flow-based image synthesis model. We further optimize the system using the publicly available HPS-v2.1 model, a human-preference-driven metric for visual quality. As demonstrated in Figure 1(c), this approach elevates the mean reward score from about 0.23 to 0.33, reflecting enhanced alignment with human aesthetic preferences.

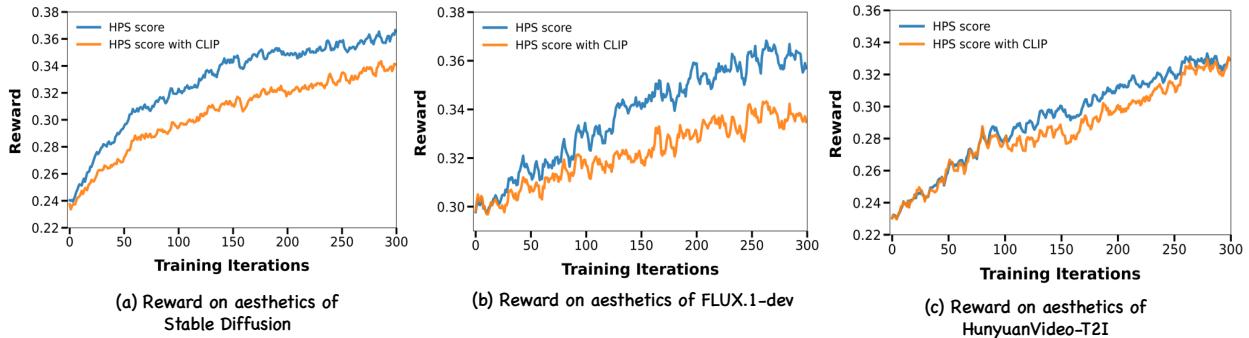


Figure 1 We visualize the reward curves of Stable Diffusion, FLUX.1-dev, and HunyuanVideo-T2I on HPS score from left to right. After applying CLIP score, the HPS score decreases, but the generated images become more natural (Figure 9 in Appendix), and the CLIP score improves (Tables 2 and 3).

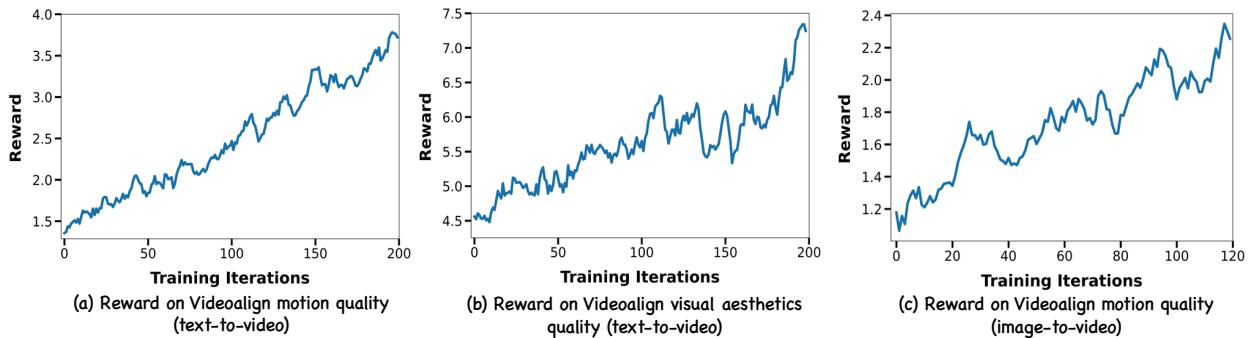


Figure 2 We visualize the training curves of motion quality and visual aesthetics quality on HunyuanVideo, motion quality on SkyReels-I2V.

3.3 Text-to-Video Generation

HunyuanVideo. Optimizing text-to-video generation models presents significantly greater challenges compared to text-to-image frameworks, primarily due to elevated computational costs during training and inference, as well as slower convergence rates. In the pretraining protocol, we always adopt a progressive strategy: initial training focuses on text-to-image generation, followed by low-resolution video synthesis, and culminates in high-resolution video refinement. However, empirical observations reveal that relying solely on image-centric

optimization leads to suboptimal video generation outcomes. To address this, our implementation employs training video samples synthesized at a resolution of 480×480 pixels, but we can visualize the samples with larger pixels.

Furthermore, constructing an effective video reward model for training alignment poses substantial difficulties. Our experiments evaluated several candidates: the Videoscore [37] model exhibited unstable reward distributions, rendering it impractical for optimization, while Visionreward-Video [38], a 29-dimensional metric, yielded semantically coherent rewards but suffered from inaccuracies across individual dimensions. Consequently, we adopted VideoAlign [11], a multidimensional framework evaluating three critical aspects: visual aesthetics quality, motion quality, and text-video alignment. Notably, the text-video alignment dimension demonstrated significant instability, prompting its exclusion from our final analysis. We also increase the number of sampled frames per second for VideoAlign to improve the training stability. As illustrated in Figure 2(a) and Figure 2(b), our methodology achieves relative improvements of 56% and **181%** in visual and motion quality metrics, respectively. Extended qualitative results are provided in the Table 5.

3.4 Image-to-Video Generation

SkyReels-I2V. SkyReels-I2V represents a state-of-the-art open-source image-to-video (I2V) generation framework, established as of February 2025 at the inception of this study. Derived from the HunyuanVideo architecture, the model is fine-tuned by integrating image conditions into the input concatenation process. A central finding of our investigation is that I2V models exclusively allow optimization of motion quality, encompassing motion coherence and aesthetic dynamics, since visual fidelity and text-video alignment are inherently constrained by the attributes of the input image rather than the parametric space of the model. Consequently, our optimization protocol leverages the motion quality metric from the VideoAlign reward model, achieving a **91%** relative improvement in this dimension as shown in Figure 2(c). We must enable the CFG-training to ensure the sampling quality during the RLHF training process.

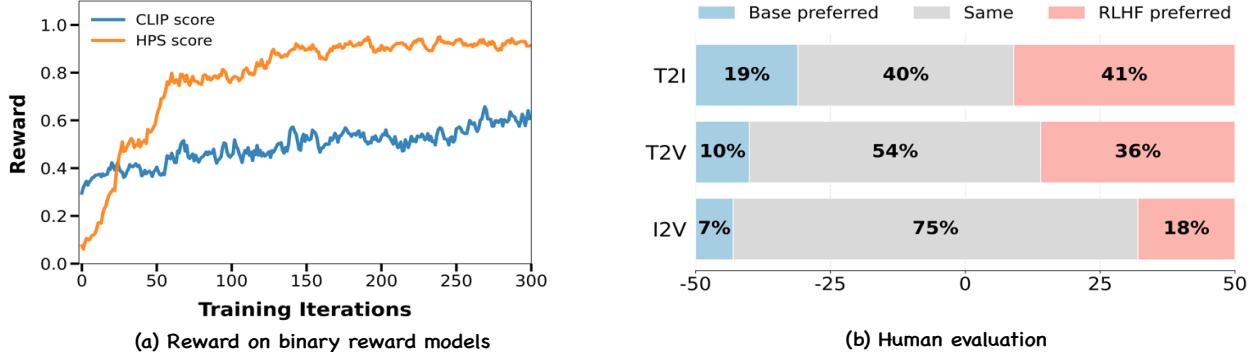


Figure 3 (a) We visualize the training curves of binary rewards. (b) We show the human evaluation results using FLUX (T2I), HunyuanVideo (T2V), and SkyReel (I2V), respectively.

3.5 Human Evaluation

We present the results of our human evaluation, conducted using in-house prompts and reference images. For text-to-image generation, we evaluate FLUX on 240 prompts. For text-to-video generation, we assess HunyuanVideo on 200 prompts, and for image-to-video generation, we test SkyReels-I2V on 200 prompts paired with their corresponding reference images. As shown in Figure 3(b), human artists consistently prefer outputs refined with RLHF. More visualization results can be found in Figure 10 in Appendix, and Appendix F.

3.6 Ablation Study

Ablation on Timestep Selection. As detailed in Section 2.2, we investigate the impact of timestep selection on the training dynamics of the HunyuanVideo-T2I model. We conduct an ablation study across three experimental conditions: (1) training exclusively on the first 30% of timesteps from noise, (2) training on

randomly sampled 30% of timesteps, (3) training on the final 40% of timesteps before outputs, (4) training on randomly sampled 60% of timesteps, and (5) training on sampled 100% of timesteps. As shown in Figure 4(b), empirical results indicate that the initial 30% of timesteps are critical for learning foundational generative patterns, as evidenced by their disproportionate contribution to model performance. However, restricting training solely to this interval leads to performance degradation compared to full-sequence training, likely due to insufficient exposure to late-stage refinement dynamics.

To reconcile this trade-off between computational efficiency and model fidelity, we always implement a 40% stochastic timestep dropout strategy during training. This approach randomly masks 40% of timesteps across all phases while preserving temporal continuity in the latent diffusion process. The findings suggest that strategic timestep subsampling can optimize resource utilization in flow-based generative frameworks.

Ablation on Noise Level ε_t . We systematically investigate the impact of noise level ε_t during training on FLUX. Our analysis reveals that reducing ε_t leads to a significant performance degradation, as quantitatively demonstrated in Figure 4(c). Notably, experiments with alternative noise decay schedules (e.g., those used in DDPM) show no statistically significant differences in output quality compared to our baseline configuration. Furthermore, the noise level larger than 0.3 sometimes leads to noisy images after RLHF training.

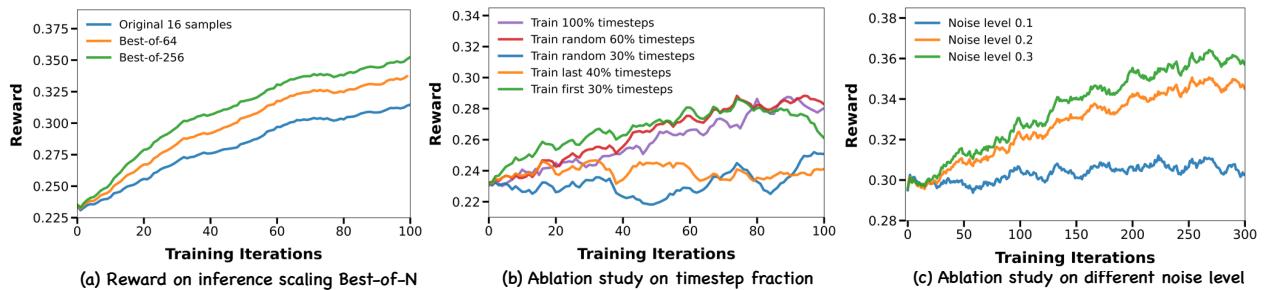


Figure 4 This figure shows the rewards on Best-of-N inference scaling, the ablation on timestep selection, and the ablation on noise level, respectively. While Best-of-N inference scaling consistently improves performance with more samples, it reduces sampling efficiency. Therefore, we leave Best-of-N as an optional extension.

4 Related Work

Aligning Large Language Models. Large Language Models (LLMs) [39–43] are typically aligned with Reinforcement Learning from Human Feedback (RLHF) [42–47]. RLHF involves training a reward function based on comparison data of model outputs to capture human preferences, which is then utilized in reinforcement learning to align the policy model. While some approaches leverage policy gradient methods, others focus on Direct Policy Optimization (DPO) [48]. Policy gradient methods have proven effective but are computationally expensive and require extensive hyperparameter tuning. In contrast, DPO offers a more cost-efficient alternative but consistently underperforms compared to policy gradient methods.

Recently, DeepSeek-R1 [16] demonstrated that the application of large-scale reinforcement learning with formatting and result-only reward functions can guide LLMs toward the self-emergence of thought processes, enabling human-like complex chain-of-thought reasoning. This approach has achieved significant advantages in complex reasoning tasks, showcasing immense potential in advancing reasoning capabilities within Large Language Models.

Aligning Diffusion Models and Rectified Flows. Diffusion models and rectified flows can also benefit significantly from alignment with human feedback, but the exploration remains primitive compared with LLMs. Key approaches in this area include: (1) Direct Policy Optimization (DPO)-style [10, 11, 38, 49, 50] methods, (2) direct backpropagation with reward signals [51], such as ReFL [9], and (3) policy gradient-based methods, including DPOK [15] and DDPO [14]. However, production-level models predominantly rely on DPO and ReFL, as previous policy gradient methods have demonstrated instability when applied to large-scale settings. Our work addresses this limitation, providing a robust solution to enhance stability and scalability. We also

hope our work offers insights into its potential to unify optimization paradigms across different modalities (e.g. image and text) [52, 53].

5 Conclusion and Future Work

This work pioneers the integration of Group Relative Policy Optimization (GRPO) into visual generation, establishing DanceGRPO as a unified framework for enhancing diffusion models and rectified flows across text-to-image, text-to-video, and image-to-video tasks. By bridging the gap between language and visual modalities, our approach addresses critical limitations of prior methods, achieving superior performance through efficient alignment with human preferences and robust scaling to complex, multi-task settings. Experiments demonstrate substantial improvements in visual fidelity, motion quality, and text-image alignment. Future work will explore GRPO’s extension to multimodal generation, further unifying optimization paradigms across Generative AI.

References

- [1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [2] 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.
- [3] 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.
- [4] Zeyue Xue, Guanglu Song, Qiushan Guo, Boxiao Liu, Zhuofan Zong, Yu Liu, and Ping Luo. Raphael: Text-to-image generation via large mixture of diffusion paths. *Advances in Neural Information Processing Systems*, 36:41693–41706, 2023.
- [5] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- [6] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.
- [7] 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.
- [8] Lixue Gong, Xiaoxia Hou, Fanshi Li, Liang Li, Xiaochen Lian, Fei Liu, Liyang Liu, Wei Liu, Wei Lu, Yichun Shi, et al. Seedream 2.0: A native chinese-english bilingual image generation foundation model. *arXiv preprint arXiv:2503.07703*, 2025.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- [13] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [14] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.
- [15] 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.
- [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] 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.

- [18] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuuanvideo: A systematic framework for large video generative models. [arXiv preprint arXiv:2412.03603](#), 2024.
- [19] Black Forest Labs. Flux. <https://github.com/black-forest-labs/flux>, 2024.
- [20] SkyReels-AI. Skyreels v1: Human-centric video foundation model. <https://github.com/SkyworkAI/SkyReels-V1>, 2025.
- [21] Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Better aligning text-to-image models with human preference. [arXiv preprint arXiv:2303.14420](#), 1(3), 2023.
- [22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In [International conference on machine learning](#), pages 8748–8763. PMLR, 2021.
- [23] 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.
- [24] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. [arXiv preprint arXiv:2010.02502](#), 2020.
- [25] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. [arXiv preprint arXiv:2409.12191](#), 2024.
- [26] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. [Advances in neural information processing systems](#), 32, 2019.
- [27] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. [arXiv preprint arXiv:2011.13456](#), 2020.
- [28] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. [arXiv preprint arXiv:2209.15571](#), 2022.
- [29] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2023. URL <https://arxiv.org/abs/2303.08797>, 3.
- [30] Ruiqi Gao, Emiel Hoogeboom, Jonathan Heek, Valentin De Bortoli, Kevin P. Murphy, and Tim Salimans. Diffusion meets flow matching: Two sides of the same coin. 2024.
- [31] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. [arXiv preprint arXiv:2207.12598](#), 2022.
- [32] Wenhao Wang and Yi Yang. Vidprom: A million-scale real prompt-gallery dataset for text-to-video diffusion models. [arXiv preprint arXiv:2403.06098](#), 2024.
- [33] Hangliang Ding, Dacheng Li, Runlong Su, Peiyuan Zhang, Zhijie Deng, Ion Stoica, and Hao Zhang. Efficient-vdit: Efficient video diffusion transformers with attention tile, 2025.
- [34] Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhenghong Liu, and Hao Zhang. Fast video generation with sliding tile attention, 2025.
- [35] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, 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.
- [36] Jiacheng Zhang, Jie Wu, Weifeng Chen, Yatai Ji, Xuefeng Xiao, Weilin Huang, and Kai Han. Onlinevpo: Align video diffusion model with online video-centric preference optimization. [arXiv preprint arXiv:2412.15159](#), 2024.
- [37] Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, et al. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. [arXiv preprint arXiv:2406.15252](#), 2024.
- [38] Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, et al. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. [arXiv preprint arXiv:2412.21059](#), 2024.

- [39] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. [arXiv preprint arXiv:2303.08774](#), 2023.
- [40] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kajie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. [ACM transactions on intelligent systems and technology](#), 15(3):1–45, 2024.
- [41] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. [Advances in neural information processing systems](#), 33:1877–1901, 2020.
- [42] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. [arXiv preprint arXiv:2407.21783](#), 2024.
- [43] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. [arXiv preprint arXiv:2412.15115](#), 2024.
- [44] 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.
- [45] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. 2023.
- [46] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In [International Conference on Machine Learning](#), pages 10835–10866. PMLR, 2023.
- [47] 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.
- [48] 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.
- [49] Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Ji Li, and Liang Zheng. Step-aware preference optimization: Aligning preference with denoising performance at each step. [arXiv preprint arXiv:2406.04314](#), 2(5):7, 2024.
- [50] Tao Zhang, Cheng Da, Kun Ding, Kun Jin, Yan Li, Tingting Gao, Di Zhang, Shimeng Xiang, and Chunhong Pan. Diffusion model as a noise-aware latent reward model for step-level preference optimization. [arXiv preprint arXiv:2502.01051](#), 2025.
- [51] Mihir Prabhudesai, Russell Mendonca, Zheyang Qin, Katerina Fragkiadaki, and Deepak Pathak. Video diffusion alignment via reward gradients. [arXiv preprint arXiv:2407.08737](#), 2024.
- [52] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Liang Zhao, et al. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. [arXiv preprint arXiv:2411.07975](#), 2024.
- [53] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In [Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition](#), pages 14398–14409, 2024.

Appendix

A Experimental Settings

We provide detailed experimental settings in Table 6, which apply exclusively to training without classifier-free guidance (CFG). When enabling CFG, we configure one gradient update per iteration. Additionally, the sampling steps vary by model: we use 50 steps for Stable Diffusion, and 25 steps for FLUX and HunyuanVideo.

Table 6 Our Hyper-paramters.

Learning rate	1e-5
Optimizer	AdamW
Gradient clip norm	1.0
Prompts per iteration	32
Images per prompt	12
Gradient updates per iteration	4
Clip range ϵ	1e-4
Noise level ε_t	0.3
Timestep Selection τ	0.6

B More Analysis

B.1 Stochastic Interpolants

The stochastic interpolant framework, introduced by [29], offers a unifying perspective on generative models like rectified flows and score-based diffusion models. It achieves this by constructing a continuous-time stochastic process that bridges any two arbitrary probability densities, ρ_0 and ρ_1 .

In our work, we connect with a specific type of stochastic interpolant known as spatially linear interpolants, defined in Section 4 of [29]. Given densities $\rho_0, \rho_1 : \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0}$, a spatially linear stochastic interpolant process x_t is defined as:

$$\mathbf{z}_t = \alpha(t)\mathbf{z}_0 + \beta(t)\mathbf{z}_1 + \gamma(t)\boldsymbol{\epsilon}, \quad t \in [0, 1], \quad (11)$$

where $\mathbf{z}_0 \sim \rho_0$, $\mathbf{z}_1 \sim \rho_1$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$ is a standard Gaussian random variable independent of \mathbf{z}_0 and \mathbf{z}_1 . The functions $\alpha, \beta, \gamma : [0, 1] \rightarrow \mathbb{R}$ are sufficiently smooth and satisfy the boundary conditions:

$$\alpha(0) = \beta(1) = 1, \quad \alpha(1) = \beta(0) = \gamma(0) = \gamma(1) = 0, \quad (12)$$

with the additional constraint that $\gamma(t) \geq 0$ for $t \in (0, 1)$. The term $\gamma(t)\boldsymbol{\epsilon}$ introduces latent noise, smoothing the path between densities.

Specific choices within this framework recover familiar models:

- **Rectified Flow (RF):** Setting $\gamma(t) = 0$ (removing the latent noise), $\alpha(t) = 1 - t$, and $\beta(t) = t$ yields the linear interpolation $\mathbf{z}_t = (1 - t)\mathbf{z}_0 + t\mathbf{z}_1$ used in Rectified Flow [6, 29]. The dynamics are typically governed by an ODE $d\mathbf{z}_t = \mathbf{u}_t dt$, where \mathbf{u}_t is the learned velocity field.
- **Score-Based Diffusion Models (SBDM):** The framework connects to SBDMs via one-sided linear interpolants (Section 4.4 of [29]), where ρ_1 is typically Gaussian. The interpolant takes the form $\mathbf{z}_t = \alpha(t)\mathbf{z}_0 + \beta(t)\mathbf{z}_1$. The VP-SDE formulation [27] corresponds to choosing $\alpha(t) = \sqrt{1 - t^2}$ and $\beta(t) = t$ after a time reparameterization.

A pivotal insight is that: "The law of the interpolant z_t at any time $t \in [0, 1]$ can be realized by many different processes, including an ODE and forward and backward SDEs whose drifts can be learned from data."

The stochastic interpolant framework provides a probability flow ODE for RF:

$$dz_t = u_t dt. \quad (13)$$

The backward SDE associated with the interpolant's density evolution is given by:

$$dz_t = b_B(t, x_t) dt + \sqrt{2\epsilon(t)} dz, \quad (14)$$

where $b_B(t, x) = u_t - \epsilon(t)s(t, x)$ is the backward drift, $s(t, x)$ is the score function, and $\epsilon(t) \geq 0$ is a tunable diffusion coefficient (noise schedule). If we set $\varepsilon_t = \sqrt{2\epsilon(t)}$, the backward SDE becomes:

$$dz_t = \left(u_t - \frac{\varepsilon_t^2}{2} \nabla \log p_t(z_t) \right) dt + \varepsilon_t dz, \quad (15)$$

which is the same as [30].

B.2 Connections between Rectified Flows and Diffusion Models

We aim to demonstrate the equivalence between certain formulations of diffusion models and flow matching (specifically, stochastic interpolants) by deriving the hyperparameters of one model from the other.

The forward process of a diffusion model is described by an SDE:

$$dz_t = f_t z_t dt + g_t dw, \quad (16)$$

where dw is a Brownian motion, and f_t, g_t define the noise schedule.

The corresponding generative (reverse) process SDE is given by:

$$dz_t = \left(f_t z_t - \frac{1 + \eta_t^2}{2} g_t^2 \nabla \log p_t(z_t) \right) dt + \eta_t g_t dw, \quad (17)$$

where $p_t(z_t)$ is the marginal probability density of z_t at time t .

For flow matching, we consider an interpolant path between data $x = z_0$ and noise ϵ (typically $\epsilon \sim \mathcal{N}(0, I)$):

$$z_t = \alpha_t x + \sigma_t \epsilon. \quad (18)$$

This path satisfies the ODE:

$$dz_t = u_t dt, \quad \text{where } u_t = \dot{\alpha}_t x + \dot{\sigma}_t \epsilon. \quad (19)$$

This can be generalized to a stochastic interpolant SDE:

$$dz_t = (u_t - \frac{1}{2} \varepsilon_t^2 \nabla \log p_t(z_t)) dt + \varepsilon_t dw. \quad (20)$$

The core idea is to match the marginal distributions $p_t(z_t)$ generated by the forward diffusion process Eq.(16) with those implied by the interpolant path Eq.(18). We will derive f_t and g_t from this requirement, and then relate the noise terms of the generative SDEs Eq.(17) and Eq.(20) to find η_t .

Deriving f_t by Matching Means. From Eq.(18), assuming $x = z_0$ is fixed and $E[\epsilon] = \mathbf{0}$, the mean of z_t is $E[z_t] = \alpha_t x$. The mean $m_t = E[z_t]$ of the process Eq.(16) starting from $z_0 = x$ satisfies the ODE $\frac{dm_t}{dt} = f_t m_t$. We require $m_t = \alpha_t x$ for all t . Substituting into the mean ODE:

$$\frac{d}{dt}(\alpha_t x) = f_t(\alpha_t x), \quad \dot{\alpha}_t x = f_t \alpha_t x. \quad (21)$$

Assuming this holds for any \mathbf{x} and $\alpha_t \neq 0$, we divide by $\alpha_t \mathbf{x}$:

$$f_t = \frac{\dot{\alpha}_t}{\alpha_t} \quad (22)$$

Using the identity $\frac{d}{dt} \log(y) = \dot{y}/y$, we get:

$$f_t = \partial_t \log(\alpha_t) \quad (23)$$

Deriving g_t^2 by Matching Variances. From Eq.(18), assuming \mathbf{x} is fixed and $Var(\epsilon) = \mathbf{I}$ (identity matrix for standard Gaussian noise), the variance (covariance matrix) of \mathbf{z}_t is $Var(\mathbf{z}_t) = Var(\alpha_t \mathbf{x} + \sigma_t \epsilon) = \sigma_t^2 Var(\epsilon) = \sigma_t^2 \mathbf{I}$. Let $V_t = \sigma_t^2$ be the scalar variance magnitude. The variance $V_t = \text{Tr}(Var(\mathbf{z}_t))/d$ for the process Eq.(16) evolves according to the Lyapunov equation: $\frac{dV_t}{dt} = 2f_t V_t + g_t^2$ (Here, g_t^2 represents the magnitude of the noise injection rate). We require $V_t = \sigma_t^2$. Substitute $V_t = \sigma_t^2$ and $f_t = \dot{\alpha}_t/\alpha_t$ into the variance evolution equation:

$$\frac{d}{dt}(\sigma_t^2) = 2 \left(\frac{\dot{\alpha}_t}{\alpha_t} \right) \sigma_t^2 + g_t^2, \quad 2\sigma_t \dot{\sigma}_t = 2 \frac{\dot{\alpha}_t}{\alpha_t} \sigma_t^2 + g_t^2. \quad (24)$$

Solving for g_t^2 :

$$g_t^2 = 2\sigma_t \dot{\sigma}_t - 2 \frac{\dot{\alpha}_t}{\alpha_t} \sigma_t^2 = \frac{2}{\alpha_t} (\alpha_t \sigma_t \dot{\sigma}_t - \dot{\alpha}_t \sigma_t^2) = \frac{2\sigma_t}{\alpha_t} (\alpha_t \dot{\sigma}_t - \dot{\alpha}_t \sigma_t) \quad (25)$$

Using the quotient rule for differentiation, $\partial_t(\sigma_t/\alpha_t) = \frac{\alpha_t \dot{\sigma}_t - \dot{\alpha}_t \sigma_t}{\alpha_t^2}$, which implies $\alpha_t \dot{\sigma}_t - \dot{\alpha}_t \sigma_t = \alpha_t^2 \partial_t(\sigma_t/\alpha_t)$. Then we get:

$$g_t^2 = \frac{2\sigma_t}{\alpha_t} \left(\alpha_t^2 \partial_t \left(\frac{\sigma_t}{\alpha_t} \right) \right) = 2\alpha_t \sigma_t \partial_t \left(\frac{\sigma_t}{\alpha_t} \right) \quad (26)$$

Thus, we have:

$$g_t^2 = 2\alpha_t \sigma_t \partial_t \left(\frac{\sigma_t}{\alpha_t} \right) \quad (27)$$

Deriving η_t by Matching Noise Terms in Generative SDEs. We compare the coefficients of the Brownian motion term ($d\mathbf{w}$) in the reverse diffusion SDE Eq.(17) and the stochastic interpolant SDE Eq.(20). The diffusion coefficient (magnitude of the noise term) is $D_{\text{diff}} = \eta_t g_t$. The diffusion coefficient is $D_{\text{int}} = \varepsilon_t$. To match the noise structure in these specific SDE forms, we set $D_{\text{diff}} = D_{\text{int}}$: $\eta_t g_t = \varepsilon_t$. Solving for η_t (assuming $g_t \neq 0$): $\eta_t = \frac{\varepsilon_t}{g_t}$. Substitute $g_t = \sqrt{g_t^2}$ using the result from Eq.(27):

$$\eta_t = \frac{\varepsilon_t}{\sqrt{2\alpha_t \sigma_t \partial_t(\sigma_t/\alpha_t)}} \quad (28)$$

Summary of Results. By requiring the forward diffusion process Eq.(16) to match the marginal mean and variance of the interpolant path Eq.(18) at all times t , we derived:

$$f_t = \partial_t \log(\alpha_t), \quad g_t^2 = 2\alpha_t \sigma_t \partial_t(\sigma_t/\alpha_t), \quad \eta_t = \varepsilon_t / (2\alpha_t \sigma_t \partial_t(\sigma_t/\alpha_t))^{1/2}. \quad (29)$$

These relationships establish the equivalence between the parameters of the two frameworks under the specified conditions.

C Classifier-Free Guidance (CFG) Training

Classifier-Free Guidance (CFG) [31] is a widely adopted technique for generating high-quality samples in conditional generative modeling. However, in our settings, integrating CFG into training pipelines introduces instability during optimization. To mitigate this, we empirically recommend disabling CFG during the sampling phase for models with high sample fidelity, such as HunyuanVideo and FLUX, as it reduces gradient oscillation while preserving output quality.

For CFG-dependent models like SkyReels-I2V and Stable Diffusion, where CFG is critical for reasonable sample quality, we identify a key instability: training exclusively on the conditional objective leads to divergent optimization trajectories. This necessitates the joint optimization of both conditional and unconditional outputs, effectively doubling VRAM consumption due to dual-network computations. Moreover, we propose reducing the frequency of parameter updates per training iteration. For instance, empirical validation shows that limiting updates to one per iteration significantly enhances training stability for SkyReels-I2V, with minimal impact on convergence rates.

D Advantages over DDPO and DPOK

Our approach differs from prior RL-based methods for text-to-image diffusion models (e.g., DDPO, DPOK) in three key aspects: (1) We employ a GRPO-style objective function, (2) we compute advantages within prompt-level groups rather than globally, (3) we ensure noise consistency across samples from the same prompt, (4) we generalize these improvements beyond diffusion models by applying them to rectified flows and scaling to video generation tasks.

E Inserting DDPO into Rectified Flow SDEs

We also insert DDPO-style objective function into rectified flow SDEs, but it always diverges, as shown in Figure 5, which demonstrates the superiority of DanceGRPO.

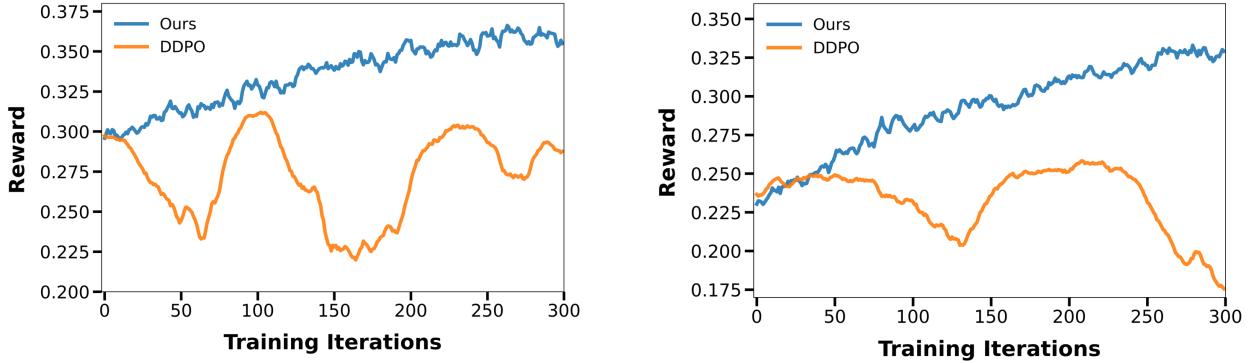


Figure 5 We visualize the results of DDPO and Ours. DDPO always diverges when applied to rectified flow SDEs.

F More Visualization Results

We provide more visualization results on FLUX, Stable Diffusion, and HunyuanVideo as shown in Figure 6, 7, 10, 11, 12, 13, 14, 15, 16, 17, and 18.



Prompt: A man lying down on green grass, gazing at the stars during an evening at a countryside villa



Prompt: A sinister man with red eyes speaking, very close shot, Cthulhu



Prompt: A chubby baby playing with toys in the snow



Prompt: Generate a picture of a blue sports car parked on the road, metal texture

Figure 6 We visualize the results by selecting FLUX optimized with the HPS score at iterations 0, 60, 120, 180, 240, and 300. The optimized outputs tend to exhibit brighter tones and richer details. However, incorporating CLIP score regularization is crucial, as demonstrated in Figures 11, 12, and 13.



Figure 7 Visualization of the diversity of the model before and after RLHF. Different seed tends to generate similar images after RLHF.



Figure 8 Visualization of the results of reward hacking (with different initialization noise) and without reward hacking (with the same initialization noise) on HunyuanVideo. Prompt: A splash of water in a clear glass, with sparkling, clear radiant reflections, sunlight, sparkle



Figure 9 This figure demonstrates the impact of the CLIP score. The prompt is "A photo of cup". We find that the model trained solely with HPS-v2.1 rewards tends to produce unnatural ("oily") outputs, while incorporating CLIP scores helps maintain more natural image characteristics.

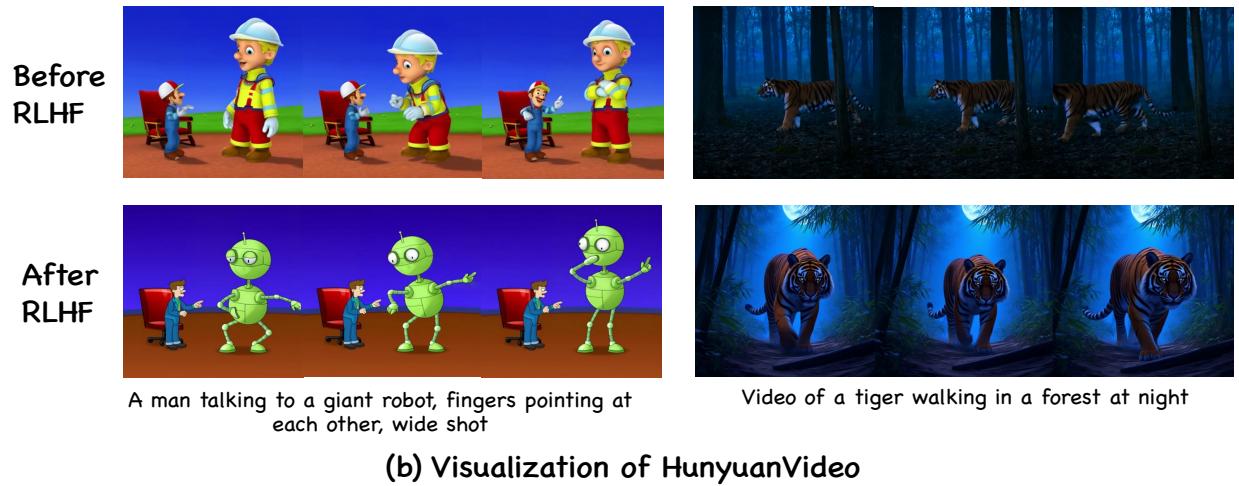
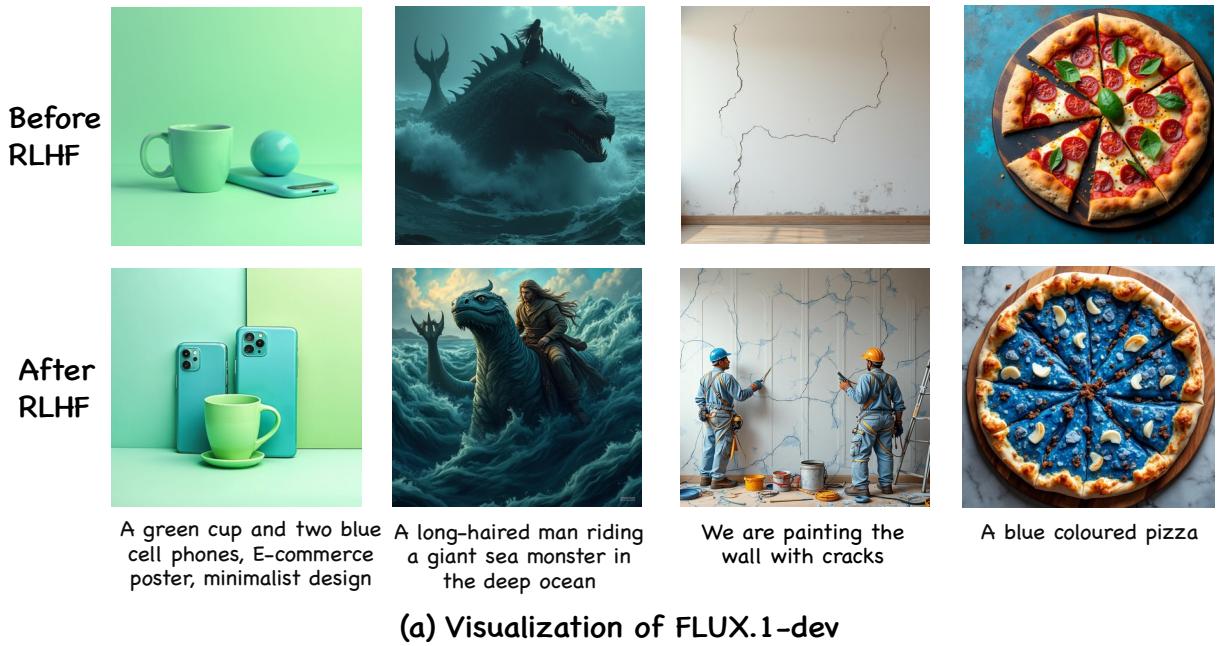


Figure 10 Overall visualization. We visualize the results before and after RLHF of FLUX and HunyuanVideo.

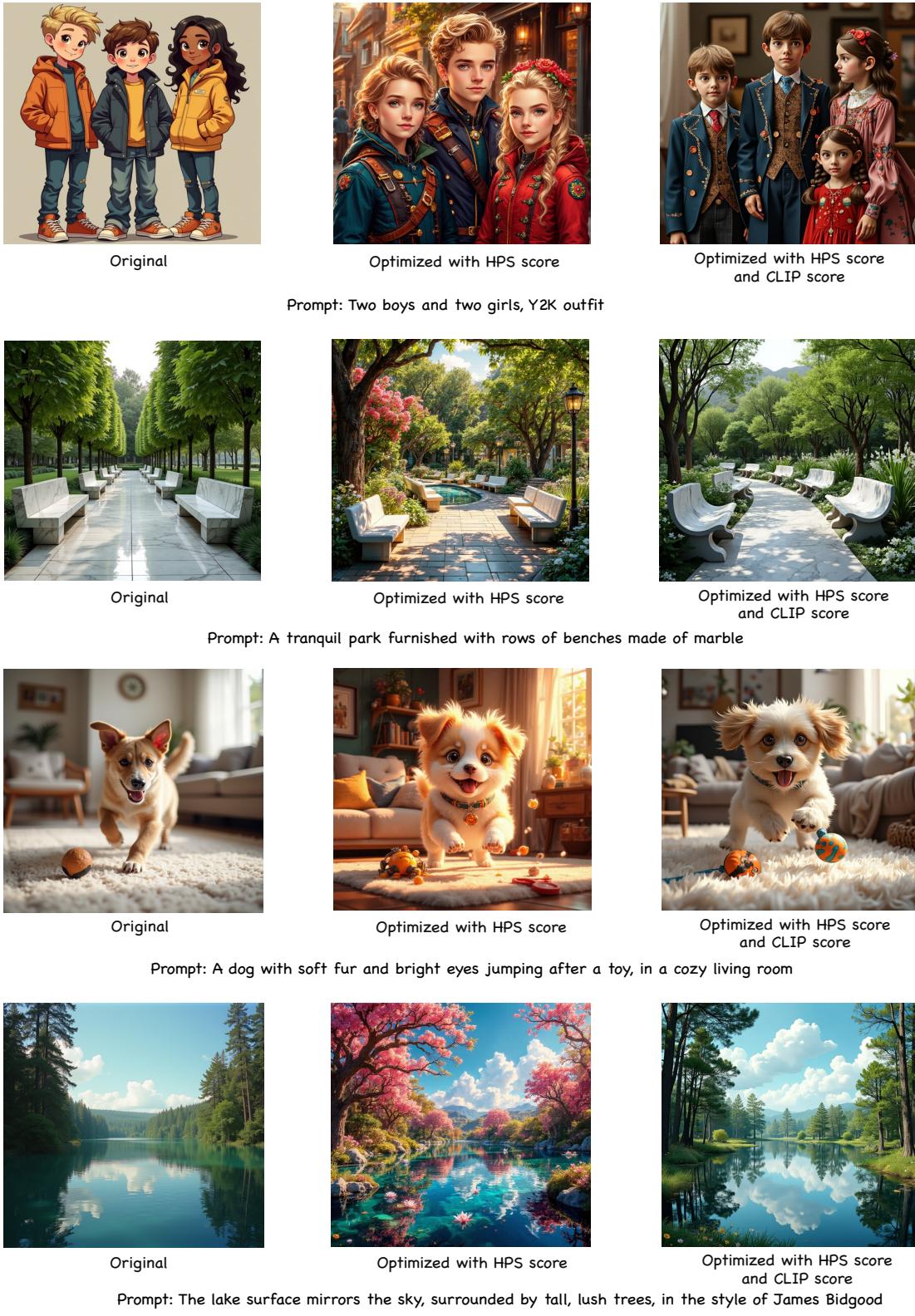
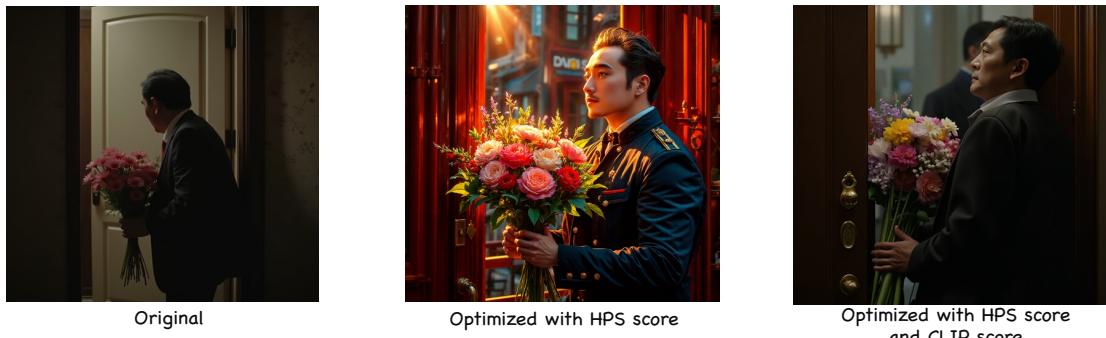


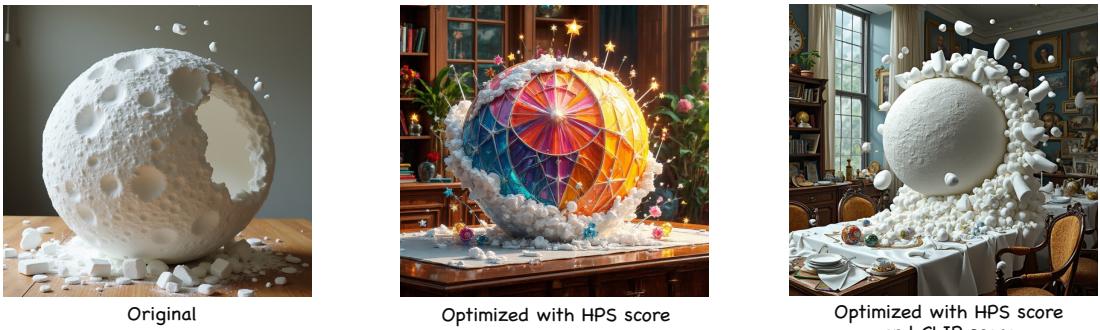
Figure 11 We present the original outputs of FLUX, alongside optimizations driven solely by the HPS score and those enhanced by both the HPS and CLIP scores.



Prompt: A photo of a soft, plush teddy bear sits on a child's bed, diagonal composition



Prompt: A man with a bunch of flowers is knocking on the door, Wong Kar-wai style, movie screenshot



Prompt: The large ball crashed right through the table because it was made of styrofoam



Prompt: A group of mongooses scuttle about, set against a desert backdrop, bathed in bright and warm earth tones

Figure 12 We present the original outputs of FLUX, alongside optimizations driven solely by the HPS score and those enhanced by both the HPS and CLIP scores.

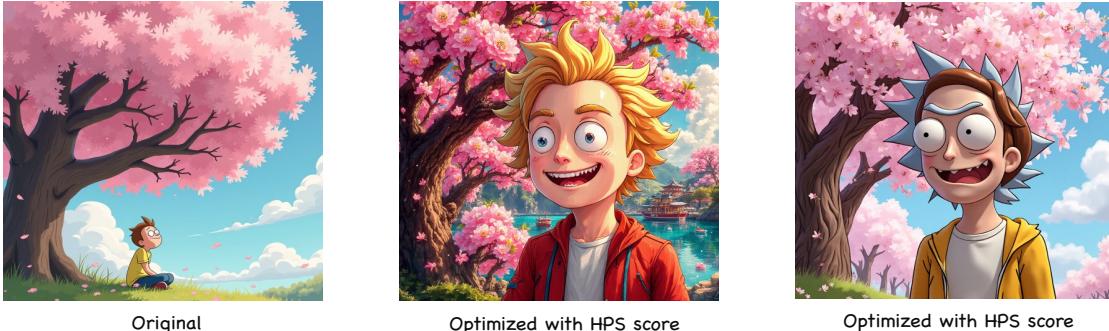


Original

Optimized with HPS score

Optimized with HPS score and CLIP score

Prompt: A television made of water that displays an image of a cityscape at night



Original

Optimized with HPS score

Optimized with HPS score and CLIP score

Prompt: A boy is smiling under the cherry blossom tree, Rick and Morty



Original

Optimized with HPS score

Optimized with HPS score and CLIP score

Prompt: A cup with a diameter of 5 cm and a cup with a diameter of 3cm



Original

Optimized with HPS score

Optimized with HPS score and CLIP score

Prompt: A family, medium shot

Figure 13 We present the original outputs of FLUX, alongside optimizations driven solely by the HPS score and those enhanced by both the HPS and CLIP scores.



Original

Optimized with HPS score

Optimized with HPS score
and CLIP score

Prompt: The feeling of a mirage in a house, the sunset shining on the lake surface, and the water flowing slowly



Original

Optimized with HPS score

Optimized with HPS score
and CLIP score

Prompt: Meat all over the place and blood and a green plant in the middle, cinematic lighting, 4k



Original

Optimized with HPS score

Optimized with HPS score
and CLIP score

Prompt: A bear is walking away in the forest, sunset



Original

Optimized with HPS score

Optimized with HPS score
and CLIP score

Prompt: Daft punk driving through the city at night

Figure 14 We present the original outputs of HunyuanVideo-T2I, alongside optimizations driven solely by the HPS score and those enhanced by both the HPS and CLIP scores.

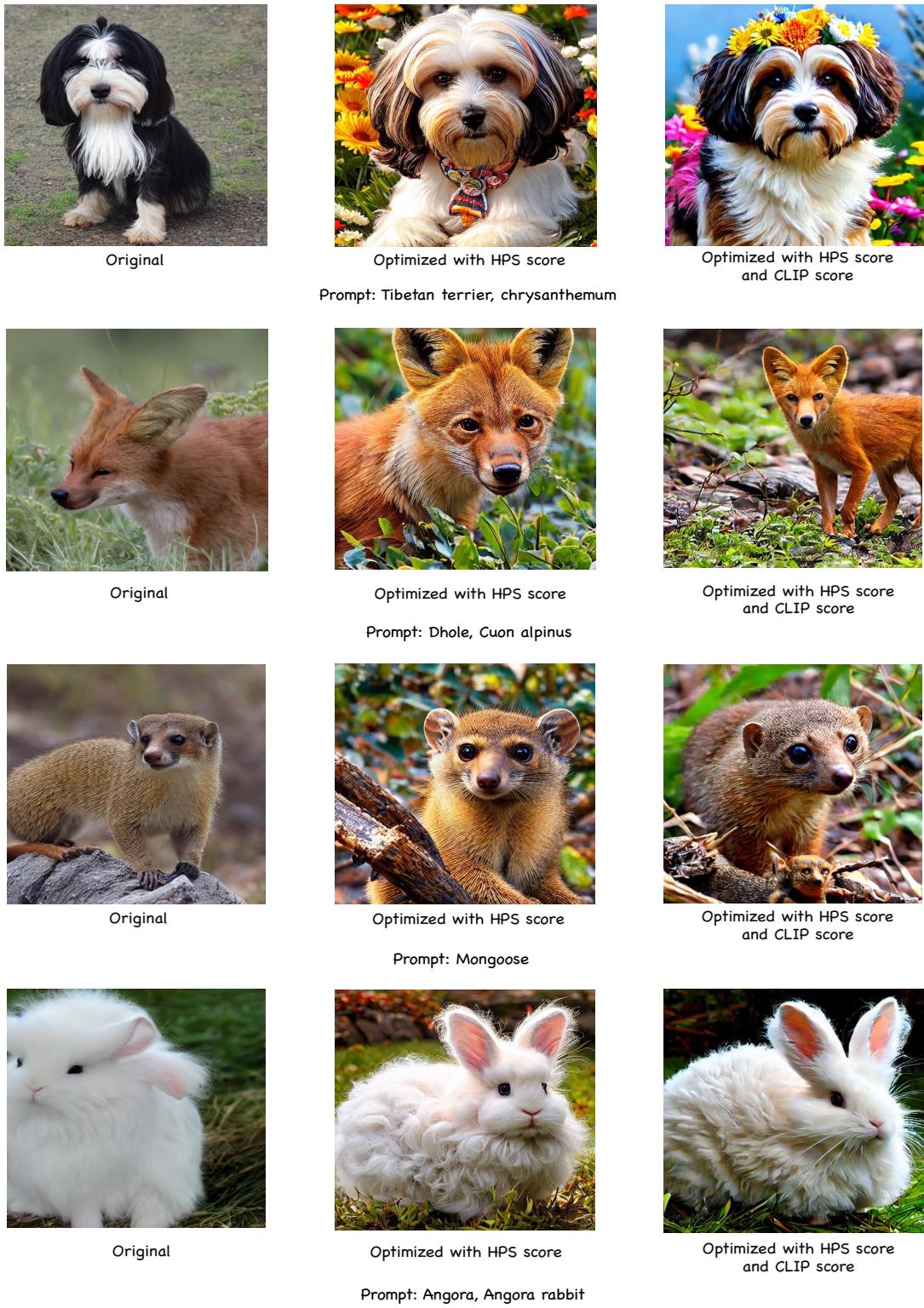


Figure 15 We present the original outputs of Stable Diffusion, alongside optimizations driven solely by the HPS score and those enhanced by both the HPS and CLIP scores.



Prompt: Dogs running with shoes 4k realism cinematic



Prompt: Tobuscus wearing a green shirt, gliding through the sky on magic shoes



Prompt: Realistic scene 4k HD angels in unique armor with gems and gold white on it with huge wings fighting among themselves

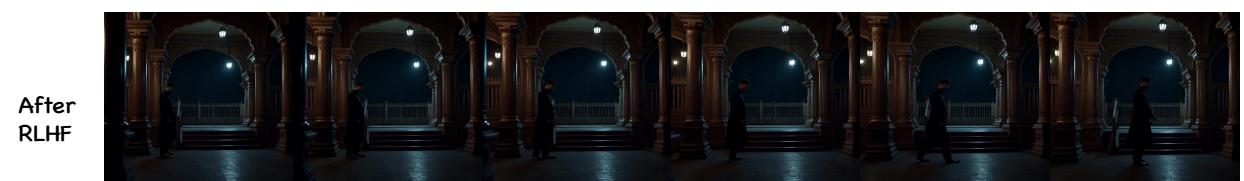
Figure 16 Visualization results of HunyuanVideo.



Prompt: Man asking the girl



Prompt: In a cold night, a girl is trying to sell the matches, hungry and tired, but still believe that it will become better



Prompt: A horror scene at night inside an Indian palace

Figure 17 Visualization results of HunyuanVideo.



Prompt: Man walking in an abandoned city in the rain



Prompt: A man leaning on a tree at the beach



Prompt: A pirate ship is sailing in the galaxy sky

Figure 18 Visualization results of HunyuanVideo.



Prompt: A black Porsche drifting in the desert



Prompt: Large light blue ice dragon breathing blue fire on town



Prompt: A young black girl walking down a street with a lot of huge trees



Prompt: a cinematic shot of a group of ultra marathon runners, 1980s, a street surrounded by fields, morning hours, anamorphic lens, Kodak Filmstock

Figure 19 Visualization results of SkyReels-I2V.