
Improving Video Generation with Human Feedback

Jie Liu^{1,3,5*} Gongye Liu^{2,3*} Jiajun Liang^{3†} Ziyang Yuan^{2,3} Xiaokun Liu³ Mingwu Zheng³
Xiele Wu^{3,4} Qiulin Wang³ Menghan Xia³ Xintao Wang³ Xiaohong Liu⁴ Fei Yang³
Pengfei Wan³ Di Zhang³ Kun Gai³ Yujiu Yang^{2✉} Wanli Ouyang^{1,5}

¹MMLab, CUHK

²Tsinghua University

³Kling Team, Kuaishou Technology

⁴Shanghai Jiao Tong University

⁵Shanghai AI Laboratory

*Equal contribution

†Project Leader

✉ Corresponding author

Project page: <https://gongyeliu.github.io/videoalign/>

Abstract

Video generation has achieved significant advances through rectified flow techniques, but issues like unsmooth motion and misalignment between videos and prompts persist. In this work, we develop a systematic pipeline that harnesses human feedback to mitigate these problems and refine the video generation model. Specifically, we begin by constructing a large-scale human preference dataset focused on modern video generation models, incorporating pairwise annotations across multi-dimensions. We then introduce VideoReward, a multi-dimensional video reward model, and examine how annotations and various design choices impact its rewarding efficacy. From a unified reinforcement learning perspective aimed at maximizing reward with KL regularization, we introduce three alignment algorithms for flow-based models. These include two training-time strategies: direct preference optimization for flow (Flow-DPO) and reward weighted regression for flow (Flow-RWR), and an inference-time technique, Flow-NRG, which applies reward guidance directly to noisy videos. Experimental results indicate that VideoReward significantly outperforms existing reward models, and Flow-DPO demonstrates superior performance compared to both Flow-RWR and supervised fine-tuning methods. Additionally, Flow-NRG lets users assign custom weights to multiple objectives during inference, meeting personalized video quality needs.

1 Introduction

Advancements in video generation have led to powerful models [57, 33, 32, 5] that produce realistic details and coherent motion. Despite this, current systems still face challenges like unstable motion, imperfect text-video alignment and insufficient alignment with human preferences [84]. In language model and image generation, reinforcement learning from human feedback (RLHF) [52, 90, 71] has proven effective in improving response quality and aligning models with user expectations.

However, applying RLHF to video generation is still remains in its infancy. A major obstacle is the lack of a reliable reward signal. Existing preference datasets [19, 73, 47, 76] were collected on earlier T2V models that produced short, low-resolution clips. Reward models trained on such data may miss fine spatial detail and long-range dynamics, while over-penalising glitches that current T2V models already suppress. In addition, the design space of VLM-based reward models remains under-explored, leading to sub-optimal annotation paradigms, reward hacking issues, and entangled multi-attribute scores. The resulting supervision is therefore noisy, biased, and easily exploited during RLHF.

A second challenge arises from the internal mechanisms of cutting-edge video generation models. Many modern systems employ rectified flow [48, 45], predicting velocity rather than noise. Recent studies [73, 85, 47, 76] have tested DPO [61, 71] and RWR [54, 36, 17] on diffusion-based video

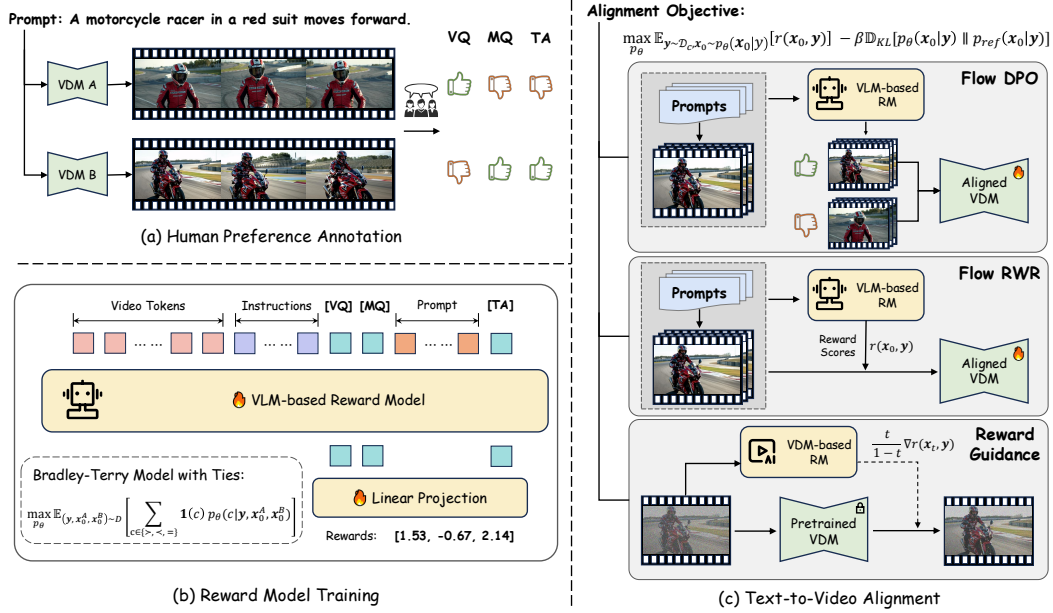


Figure 1: **Overview of Video Alignment Framework.** (a) **Human Preference Annotation.** We construct 182k prompt-video triplets, each annotated on Visual Quality (VQ), Motion Quality (MQ), and Text Alignment (TA). (b) **Reward Modeling.** A VLM-based reward model is trained under the Bradley-Terry-Model-with-Ties formulation. (c) **Video Alignment.** From a unified RL perspective, we introduce three alignment algorithms for flow-based video generation: Flow-DPO, Flow-RWR, and Reward Guidance (Flow-NRG), and provide a systematic comparison.

generation approaches. However, adapting existing alignment methods to flow-based models introduces new questions. A recent attempts [13] for flow matching based DPO even degrade quality compared with the unaligned baseline.

To address these challenges, we present a comprehensive investigation into aligning advanced flow-based video generation models with human preferences, as shown in Fig. 1. We first collect 16k high-quality prompts and render them with 12 representative T2V models, producing 182k annotated examples across three key dimensions: Visual Quality (VQ), Motion Quality (MQ), and Text Alignment (TA). We then develop a multi-dimensional video reward model, systematically analyzing how different annotations and design choices affect its performance. These dimensions can be aggregated into a total reward that reflects the overall preferences of humans.

Armed with a reliable reward model, we revisit RLHF algorithms for rectified flow. From a unified reinforcement learning perspective that maximizes reward with KL regularization, we derive two training-time strategies—Flow-DPO and Flow-RWR, and an inference-time technique, Flow-NRG. We discover that a simple extension of Diffusion-DPO performs poorly because its timestep-dependent KL term pushes the policy to overfit the objective at higher noise levels, making the model vulnerable to reward hacking. In contrast, our Flow-DPO removes this time-dependent term and keeps the KL weight constant, showing robust performance across all dimensions. Flow-NRG is an efficient inference-time alignment algorithm that applies reward guidance directly to noisy videos during the denoising process. It allows users to apply arbitrary weightings to multiple alignment objectives during inference, eliminating the need for extensive retraining to meet personalized user requirements.

Our contributions can be summarized as follows:

- **Large-Scale Preference Dataset:** We create a 182k-sized, multi-dimensional, human-labeled video generation preference dataset from 12 modern video generation models.
- **Multi-Dimensional Reward Modeling:** We propose and systematically study a multi-dimensional video reward model, investigating how different design decisions influence its rewarding efficacy.

- **VideoGen-RewardBench:** We create a benchmark for modern reward models by annotating prompt-video pairs from VideoGen-Eval. This dataset, consisting of diverse prompts and videos generated by modern VDMs, results in 26.5k annotated video pairs with preference labels.
- **Flow-Based Alignment:** From a unified RL perspective, we introduce two training-time alignment strategies (Flow-DPO, Flow-RWR) and one inference-time technique (Flow-NRG). Experiments show that Flow-DPO with a fixed KL term outperforms other methods. And Flow-NRG allows custom weightings of multiple alignment objectives during inference.

2 Related Work

Reward modeling [75, 77, 31, 86, 41] trains CLIP-based models on human preference datasets, while newer approaches use VLMs with regression heads to predict multi-dimensional scores. Learning paradigms include point-wise regression [19, 76] and pair-wise comparison via Bradley-Terry loss. However, most video reward models focus on short, low-quality videos from the pre-Sora era [51] and lack rigorous evaluation of design choices. We address these limitations by targeting modern video generation and exploring broader reward modeling strategies. For alignment, image generation has adopted RLHF-style techniques such as reward backpropagation [58, 77], RWR [36], DPO [61, 71], PPO [2, 16], and training-free methods [80]. Concurrent efforts [17, 47, 76, 85, 73] extend DPO/RWR to diffusion-based video models using old generation models or image-level rewards. We build on this by extending DPO to flow-based video generation, proposing Flow-DPO. More comprehensive discussions of related work can be found in Appendix A.

3 VideoReward

Robust RLHF begins with a reward model that faithfully mirrors human preferences, yet existing efforts are limited in two key respects: (1) **Data:** existing video-preference datasets were curated for earlier T2V models and mismatch with what users prefer in modern video generation models; (2) **Model design:** the key technique choices for VLM-based reward models remain largely uncharted. Fig. 1 (a), (b) summarises our solution. We first build a large-scale preference dataset collected with state-of-the-art T2V models; then perform a systematic study of reward-modeling design.

3.1 Human Preference Data Collection

Existing human preference datasets for video generation [49, 26, 19, 73, 76] were primarily built on early, low-resolution T2V models that produced short, artifact-laden clips. As VDMs continue to evolve, modern T2V models, however, generate longer, higher-fidelity videos with smoother motion. Consequently, legacy datasets no longer accurately reflect what users prefer today. Reward models trained on such collections may miss fine spatial details and long-range dynamics while over-weighting temporal glitches already mitigated by current models. To bridge this gap, we develop a new preference dataset expressly for state-of-the-art VDMs.

Table 1: Statistics of the collected training dataset. We utilize 12 T2V models to generate 108k videos from 16k unique prompts, resulting in 182k annotated triplets. Each triplet consists of a prompt paired with two videos, and corresponding preference annotations.

	T2V Model	Date	#Videos	#Anno Triplets	Resolution	Duration
<i>Pre-Sora-Era Models</i>	Gen2 [63]	23.06	6k	13k	768 × 1408	4s
	SVD [3]	23.11	6k	13k	576 × 1024	4s
	Pika 1.0 [34]	23.12	6k	13k	720 × 1280	3s
	Vega [70]	23.12	6k	13k	576 × 1024	4s
	PixVerse v1 [56]	24.01	6k	13k	768 × 1408	4s
	HiDream [21]	24.01	0.3k	0.3k	768 × 1344	5s
<i>Modern Models</i>	Dreamina [6]	24.03	16k	68k	720 × 1280	6s
	Luma [50]	24.06	16k	57k	752 × 1360	5s
	Gen3 [64]	24.06	16k	55k	768 × 1280	5s
	Kling 1.0 [33]	24.06	6k	33k	384 × 672	5s
	PixVerse v2 [56]	24.07	16k	58k	576 × 1024	5s
	Kling 1.5 [33]	24.09	7k	28k	704 × 1280	5s

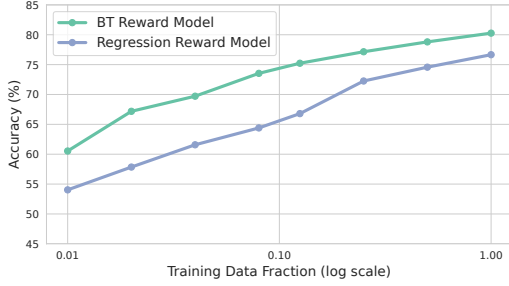


Figure 2: **BT vs. Regression.** Accuracy curves across log-scaled data fractions.

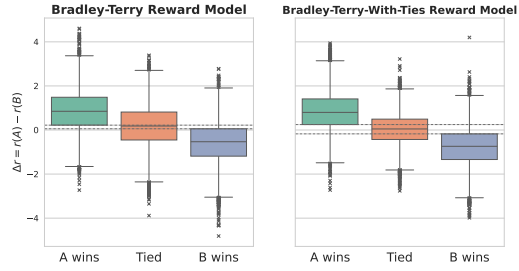


Figure 3: **BT vs. BTT.** Score-difference distributions (Δr) for BT (left) and BTT (right).

Prompt Collection and Video Generation. We collect diverse prompts from the Internet, categorize them into 8 meta-categories—*animal*, *architecture*, *food*, *people*, *plants*, *scenes*, *vehicles*, *objects*—and expand them with GPT-4o. After removing repetitive, irrelevant, or unsafe entries, we refine them with our in-house prompt rewriter, yielding **16000 high-quality prompts**. 12 T2V models of varying capabilities then render these prompts into **108k videos**, which we pair to form **182k triplets**, each comprising a prompt and two corresponding videos from distinct VDMs. Comprehensive dataset statistics are provided in Tab. 1 and Fig. 8.

Multi-dimensional Annotation. Professional annotators were hired to view each triplet and record pairwise preferences (*A wins* / *Ties* / *B wins*) separately for *Visual Quality (VQ)*, *Motion Quality (MQ)*, and *Text Alignment (TA)*, producing a three-label vector per triplet. The same annotators also assign 1–5 Likert scores to each individual video, enabling later studies that compare pointwise and pairwise supervision. We reserve **13 000** triplets whose prompts never appear in training as a validation set. Detailed annotation protocols are provided in Appendix G.

3.2 Reward Modeling

Prior works [19, 73, 76] on VLM-based reward models have proved effective for both evaluation [75, 19] and optimization [36, 71, 59]. However, their core design choices remain insufficiently explored, and current methods still suffer from issues like annotation paradigms, reward hacking and entangled multi-attribute signals. We select the lightweight Qwen2-VL-2B [72] as our backbone, and systematically examine three key designs and demonstrate how each yield a cleaner, more reliable reward signal for RLHF. We also conduct a comprehensive ablation study on the evaluation benchmark in Table 6 of Appendix E to further validate the key design choices.

Score Regression v.s. Bradley-Terry. We first investigate two reward learning paradigms: the *Bradley-Terry* (BT) model [4] and pointwise score regression. The BT model formulates preference learning as a probabilistic ranking task. Given a prompt $\mathbf{y}$ and paired videos $(\mathbf{x}_0^w, \mathbf{x}_0^l)$, it optimizes $\mathcal{L}_{BT} = -\mathbb{E} [\log (\sigma (r(\mathbf{x}_0^w, \mathbf{y}) - r(\mathbf{x}_0^l, \mathbf{y})))]$, where the expectation is taken over $(\mathbf{y}, \mathbf{x}_0^w, \mathbf{x}_0^l) \sim \mathcal{D}$.

In contrast, score regression directly predicts a scalar quality score $z \in \mathbb{R}$ using the MSE loss: $\mathcal{L}_{reg} = \mathbb{E} [\|r(\mathbf{x}_0, \mathbf{y}) - z\|^2]$, where the expectation is taken over $(\mathbf{y}, \mathbf{x}_0, z) \sim \mathcal{D}$. Since our training dataset includes both pointwise scores and pairwise preferences from the same annotators, we can directly compare between the two annotation paradigms. We train both types of reward models on increasing subsets of the training set and report the best validation accuracy averaged over VQ, MQ, and TA. Fig. 2 presents these results.

As the dataset grows, both the BT and regression-style models improve in accuracy, while the BT model remains consistently superior. This advantage stems from the nature of pairwise annotations, which capture subtle relative distinctions more effectively. Even when two videos receive identical pointwise scores, annotators can still differentiate subtle quality differences.

Ties Matters. While vanilla BT model is widely used to capture human preferences from chosen-rejected pairs, the importance of tie annotations is often overlooked. Inspired by recent work [46], we adopt *Bradley-Terry model with ties* (BTT) [62], an extension that accounts for tied preferences.

Formally, BTT defines a tripartite preference distribution:

$$P_\theta(c|\mathbf{y}, \mathbf{x}_0^A, \mathbf{x}_0^B) = \begin{cases} \frac{(\theta^2 - 1) \exp(r_A) \exp(r_B)}{(\exp(r_A) + \theta \exp(r_B)) (\theta \exp(r_A) + \exp(r_B))} & \text{Tie } (x_0^A = x_0^B), \\ \frac{\exp(r_A)}{\exp(r_B) + \theta \exp(r_A)} & \text{A preferred } (x_0^A \succ x_0^B), \\ \frac{\exp(r_B)}{\theta \exp(r_A) + \exp(r_B)} & \text{B preferred } (x_0^B \succ x_0^A). \end{cases} \quad (1)$$

where c denotes preference choice, $\theta > 1$ controls the tendency toward ties, with a larger θ increasing tie probability. We empirically set $\theta = 5.0$ and train the BTT model by minimizing negative log-likelihood:

$$\mathcal{L}_{\text{BTT}} = -\mathbb{E}_{(\mathbf{y}, \mathbf{x}_0^A, \mathbf{x}_0^B) \sim \mathcal{D}} \left[\sum_{c \in \{\succ, \prec, =\}} \mathbb{1}(c) \log P_\theta(c|\mathbf{y}, \mathbf{x}_0^A, \mathbf{x}_0^B) \right] \quad (2)$$

We train BT and BTT reward models under identical settings and visualize $\Delta r = r(\mathbf{x}_0^A, \mathbf{y}) - r(\mathbf{x}_0^B, \mathbf{y})$ on the validation set (Fig. 3). Although the BT model handles chosen/rejected pairs well, it struggles to handle ties—often assigning sizeable Δr to many tie pairs, conflating them with clear preferences. By contrast, BTT learns a flexible decision boundary that clusters ties near zero while preserving large margins for decisive wins and losses, yielding more reliable feedback for downstream RLHF.

Token Positioning. A common approach in LLM / MLLM-based reward modeling [52, 69, 19] attaches a linear projection head to the final token to predict multi-dimensional scores. This method forcing prompt-independent and prompt-dependent cues into one vector, causing *context leakage*: the same video can receive different visual-quality scores when paired with different prompts. We eliminate this entanglement with a simple token-positioning strategy. As shown in Fig. 1(b), two context-agnostic tokens, [VQ], [MQ], are inserted immediately after the video and before the prompt, so they can attend only to visual content. A context-aware token, [TA], is placed after the full prompt, allowing it to attend to both the video and the text. The final-layer embeddings of these tokens are then mapped to dimension-specific scores via a shared linear layer. This design removes context leakage, stabilizes visual and motion assessments, and maintains parameter efficiency. The full input template is provided in Appendix K.

4 Video Alignment

With a high-fidelity reward model established, we next introduce three alignment methods for flow-based generation models under a unified RL objective: two training-time algorithms—**Flow-DPO** and **Flow-RWR**, and one inference-time guidance technique, **Flow-NRG** (Fig. 1 (c)).

4.1 Preliminaries

Rectified Flow. Let $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ denote a data sample from the real data distribution, and $\mathbf{x}_1 \sim p(\mathbf{x}_1)$ denote a noise sample, where $\mathbf{x}_0, \mathbf{x}_1 \in \mathbb{R}^d$. Recent advanced image [15] and video generation models [57, 32] adopt the Rectified Flow [48], which defines the “noisy” data $\mathbf{x}_t$ as $\mathbf{x}_t = (1 - t) \mathbf{x}_0 + t \mathbf{x}_1$, where $t \in [0, 1]$. Then we can train a transformer model to regress the velocity field $\mathbf{v}_\theta(\mathbf{x}_t, t)$ by minimizing the Flow Matching objective [45, 48]:

$$\mathcal{L}(\theta) = \mathbb{E}_{t, \mathbf{x}_0 \sim q(\mathbf{x}_0), \mathbf{x}_1 \sim p(\mathbf{x}_1)} [\|\mathbf{v} - \mathbf{v}_\theta(\mathbf{x}_t, t)\|^2],$$

where the target velocity field is $\mathbf{v} = \mathbf{x}_1 - \mathbf{x}_0$.

RLHF. The goal of RLHF is to learn a conditional distribution $p_\theta(\mathbf{x}_0 | \mathbf{y})$ that maximizes the reward $r(\mathbf{x}_0, \mathbf{y})$ while controls the KL-divergence from the reference model p_{ref} via a coefficient β :

$$\max_{p_\theta} \mathbb{E}_{\mathbf{y} \sim \mathcal{D}_e, \mathbf{x}_0 \sim p_\theta(\mathbf{x}_0 | \mathbf{y})} [r(\mathbf{x}_0, \mathbf{y})] - \beta \mathbb{D}_{\text{KL}} [p_\theta(\mathbf{x}_0 | \mathbf{y}) \| p_{\text{ref}}(\mathbf{x}_0 | \mathbf{y})]. \quad (3)$$

4.2 Flow-DPO

Consider a training set $\mathcal{D} = \{y, x_0^w, x_0^l\}$, where y is the prompt and human annotations prefer generated video x_0^w to x_0^l (i.e., $x_0^w \succ x_0^l$). Direct Preference Optimization (DPO) [61] aligns models with human preferences by analytically solving the RLHF objective (Eq. 3) and optimizing the policy via supervised learning. Extending this idea to diffusion models, Diffusion-DPO [71] derives a DPO-style loss under the diffusion paradigm. The resulting objective $\mathcal{L}_{\text{DD}}(\theta)$ is formulated as:

$$-\mathbb{E} \left[\log \sigma \left(-\frac{\beta}{2} \left(\|\epsilon^w - \epsilon_\theta(x_t^w, t)\|^2 - \|\epsilon^w - \epsilon_{\text{ref}}(x_t^w, t)\|^2 - (\|\epsilon^l - \epsilon_\theta(x_t^l, t)\|^2 - \|\epsilon^l - \epsilon_{\text{ref}}(x_t^l, t)\|^2) \right) \right) \right] \quad (4)$$

The expectation is taken over samples $\{x_0^w, x_0^l\} \sim \mathcal{D}$ and the noise schedule t . In Rectified Flow, we relate the noise vector ϵ^* to a velocity field v^* . Specifically, Lemma B.1 in Appendix B shows that

$$\|\epsilon^* - \epsilon_{\text{pred}}(x_t^*, t)\|^2 = (1-t)^2 \|v^* - v_{\text{pred}}(x_t^*, t)\|^2, \quad (5)$$

By substituting Eq. 5 into Eq. 4, we obtain the final Flow-DPO loss $\mathcal{L}_{\text{FD}}(\theta)$:

$$-\mathbb{E} \left[\log \sigma \left(-\frac{\beta_t}{2} \left(\|v^w - v_\theta(x_t^w, t)\|^2 - \|v^w - v_{\text{ref}}(x_t^w, t)\|^2 - (\|v^l - v_\theta(x_t^l, t)\|^2 - \|v^l - v_{\text{ref}}(x_t^l, t)\|^2) \right) \right) \right] \quad (6)$$

where $\beta_t = \beta(1-t)^2$. Intuitively, minimizing $\mathcal{L}_{\text{FD}}(\theta)$ guides the predicted velocity field v_θ closer to the target velocity v^w of the “preferred” data, while pushing it away from v^l (the “less preferred” data). The strength of this preference signal depends on the differences between the predicted errors and the corresponding reference errors. We provide Flow-DPO pseudo-code in Appendix C.

Discussion on β_t . The KL coefficient β_t controls how far the learned policy is allowed to deviate from the reference model [61, 71]. A direct derivation yields the schedule $\beta_t = \beta(1-t)^2$. The penalty β_t vanishes as t approaches 1 and reaches β at $t = 0$. *This scheduling strategy causes the model to prioritize alignment at higher noise levels.* Unlike in diffusion models, our experiments reveal that this schedule degrades alignment performance in rectified flow, leading to reward hacking and visual artifacts. Inspired by a similar observation in DDPM’s [22] training objective, where discarding the weighting in denoising score matching improves sample quality, we instead adopt a constant β . This adjustment leads to more stable training and improved alignment across all reward dimensions. We provide a more detailed discussion of this in Section 5.2.

4.3 Flow-RWR

Drawing inspiration from the application of Reward-weighted Regression (RWR) [55] in diffusion models [36, 17], we propose a counterpart for flow-based models based on expectation-maximization [10]. Starting from the general KL-regularized reward-maximization problem in Eq. 3, prior work [61] shows that its optimal closed-form solution can be written as:

$$p_\theta(x_0 | y) = \frac{1}{Z(y)} p_{\text{ref}}(x_0 | y) \exp\left(\frac{1}{\beta} r(x_0, y)\right), \quad (7)$$

where $Z(y) = \sum_{x_0} p_{\text{ref}}(x_0 | y) \exp\left(\frac{1}{\beta} r(x_0, y)\right)$ is the partition function. Following [17], we can obtain the RWR loss:

$$\mathcal{L}_{\text{RWR}}(\theta) = \mathbb{E}_{y, x_0, \epsilon, t} [\exp(r(x_0, y)) \|\epsilon - \epsilon_\theta(x_t, t, y)\|^2]. \quad (8)$$

For rectified-flow models, we formulate this as a reward-weighted velocity regression:

$$\mathcal{L}_{\text{RWR}}(\theta) = \mathbb{E} [\exp(r(x_0, y)) \|v - v_\theta(x_t, t, y)\|^2], \quad (9)$$

As in Flow-DPO, we omit the $(1-t)^2$ factor for better performance.

Table 2: Preference accuracy on GenAI-Bench and VideoGen-RewardBench. For ties-excluded accuracy, we calculate accuracy using only the data labeled as “A wins” or “B wins”. For ties-included accuracy, we use the algorithm from Deutsch et al. [11], which tests various tie thresholds and selects the one that maximizes three-class accuracy. **Bold**: best performance.

Method	GenAI-Bench		VideoGen-RewardBench							
	Overall Accuracy		Overall Accuracy		VQ Accuracy		MQ Accuracy		TA Accuracy	
	w/ Ties	w/o Ties	w/ Ties	w/o Ties	w/ Ties	w/o Ties	w/ Ties	w/o Ties	w/ Ties	w/o Ties
Random	33.67	49.84	41.86	50.30	47.42	49.86	59.07	49.64	37.25	50.40
VideoScore [19]	49.03	71.69	41.80	50.22	47.41	47.72	59.05	51.09	37.24	50.34
LiFT [73]	37.06	58.39	39.08	57.26	47.53	55.97	59.04	54.91	33.79	55.43
VisionReward [76]	51.56	72.41	56.77	67.59	47.43	59.03	59.03	60.98	46.56	61.15
Ours	49.41	72.89	61.26	73.59	59.68	75.66	66.03	74.70	53.80	72.20

4.4 Noisy Reward Guidance

Recall that the KL-regularized RL objective (Eq. 3) admits a closed-form solution (Eq. 7), which transform the original distribution $p_{\text{ref}}(\mathbf{x}_0 | \mathbf{y})$ into the new target distribution $p_{\theta}(\mathbf{x}_0 | \mathbf{y})$. Since the constants β and w can be absorbed into $r(\mathbf{x}_0, \mathbf{y})$, the closed-form solution becomes:

$$p_{\theta}(\mathbf{x}_0 | \mathbf{y}) \propto p_{\text{ref}}(\mathbf{x}_0 | \mathbf{y}) [\exp(r(\mathbf{x}_0, \mathbf{y}))]^w, \quad (10)$$

where $w \in \mathbb{R}$ controls the strength of the reward guidance. For rectified flow, as we proved in Appendix B.2, this reweighting can be achieved by shifting the velocity field:

$$\tilde{\mathbf{v}}_t(\mathbf{x}_t | \mathbf{y}) = \mathbf{v}_t(\mathbf{x}_t | \mathbf{y}) - w \frac{t}{1-t} \nabla r(\mathbf{x}_t, \mathbf{y}), \quad (11)$$

This modification of the marginal velocity field alters the sampling distribution to match the target form in Eq. equation 10. Since this formulation is structurally similar to classifier guidance [12, 67], we refer to it as *reward guidance*. Pseudo-code is provided in Appendix C.

Efficient Reward on Noisy Latents. Computing ∇r in pixel space requires back-propagating through the full VAE decoder, which is computationally expensive. To address this, we propose training a lightweight, **time-dependent reward model** $r_{\theta}(\cdot, t)$ directly in latent space. For each preference pair $(\mathbf{x}^w, \mathbf{x}^l)$, we apply identical noise to both videos, assuming their relative preference remains unchanged. We then adopt the Bradley–Terry loss to learn the reward function from these noised videos. Leveraging the fact that modern VDMs are already well trained on noisy latents, we can reuse a few early layers from the pretrained backbone to construct the reward model, avoiding the need for heavy retraining. We apply Eq. equation 11 at each inference step (except at $t = 1$), enabling efficient inference-time alignment in latent space.

5 Experiments

5.1 Reward Learning

Training Setting. We use Qwen2-VL-2B [72] as the backbone of our reward model, trained with BTT loss. Several observations were made during training. First, higher video resolution and more frames generally improved the reward model’s performance. Second, using a stable sampling interval instead of a fixed frame number significantly enhanced motion quality evaluations, especially for videos of varying lengths. In practice, we sample videos at 2 fps, with a resolution of approximately 448×448 pixels while preserving the original aspect ratio. Hyperparameters are in Appendix I.

Evaluation. We evaluate our reward model on two benchmarks targeting different generations of T2V models: (1) **VideoGen-RewardBench**: Built upon the third-party prompt-video dataset VideoGen-Eval [84], this benchmark targets *modern T2V models*. We address the lack of human annotations in VideoGen-Eval by manually constructing 26.5k triplets and hiring annotators to provide pairwise preference labels. Annotators also assess overall video quality, serving as a universal label across all dimensions. (2) **GenAI-Bench** [29]: GenAI-Bench features short (2-seconds) videos generated by *pre-Sora-era T2V models*, enabling evaluation on earlier-generation outputs. A detailed comparison between the two benchmarks is provided in Appendix H.1. We evaluate our reward model against existing baselines, including VideoScore [19], as well as two concurrent works: LiFT [73]

Table 3: Multi-dimensional alignment with VQ:MQ:TA = 1:1:1. **Bold**: Best performance. Although Flow-DPO with a timestep-dependent β achieves high VQ and MQ reward win rates, it exhibits significant reward hacking. In contrast, Flow-DPO with a constant β achieves high VQ, MQ, and TA scores while avoiding reward hacking.

Method	VBench						VideoGen-Eval			TA-Hard		
	Total	Quality	Semantic	VQ	MQ	TA	VQ	MQ	TA	VQ	MQ	TA
Pretrained	83.19	84.37	78.46	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0	50.0
SFT	82.31	83.13	79.04	51.28	65.21	52.84	61.27	76.13	46.35	57.75	76.06	57.75
Flow-RWR	82.27	83.19	78.59	51.55	63.9	53.43	59.05	69.7	48.35	61.97	78.87	55.71
Flow-DPO ($\beta_t = \beta(1 - t)^2$)	80.90	81.52	78.42	87.78	82.36	51.02	88.44	91.23	28.14	84.29	83.10	38.03
Flow-DPO	83.41	84.19	80.26	93.42	69.08	75.43	90.95	81.01	68.26	77.46	71.43	73.24

Table 4: Single-dimensional alignment with TA. **Bold**: Best performance. Flow-DPO with a constant β is the most effective method, achieving best performance without reward hacking.

Method	VBench				VideoGen-Eval	TA-Hard
	Total	Quality	Semantic	TA	TA	TA
Pretrained	83.19	84.37	78.46	50.00	50.00	50.00
SFT	82.71	83.48	79.62	52.88	53.81	64.79
Flow-RWR	82.40	83.36	78.58	59.66	49.50	66.20
Flow-DPO ($\beta_t = \beta(1 - t)^2$)	82.35	83.00	79.75	63.67	55.95	71.83
Flow-DPO	83.38	84.28	79.80	69.09	65.49	84.51

and VisionReward [76]. Consistent with practices in LLM evaluation [35], we use pairwise accuracy, reporting both ties-included [11] and ties-excluded accuracy. We calculate overall accuracy on GenAI-Bench and dimension-specific (VQ, MQ, TA) accuracy on VideoGen-RewardBench. Additional evaluation details are in Appendix H.2 and Appendix H.3

Main Results. Tab. 2 presents the pairwise accuracy across both benchmarks. **VideoScore** performs well on GenAI-Bench but fails on VideoGen-RewardBench, indicating poor generalization to modern T2V models. **LiFT** improves over VideoScore on modern videos but remains below 60% accuracy, showing limited pairwise discrimination ability. **VisionReward** demonstrates competitive performance on GenAI-Bench but underperforms on VideoGen-RewardBench, especially on visual and motion dimensions under ties-included settings. This drop stems from its difficulty in assessing the improved fidelity and motion smoothness of modern outputs. In contrast, our method **VideoReward** outperforms all other models on VideoGen-RewardBench, showcasing its strong alignment with human preferences on modern T2V generations. Moreover, despite being trained on a disjoint dataset (see Fig. 10), it still achieves comparable performance on GenAI-Bench, indicating robust generalization across different eras of T2V models. **Ablation studies are provided in Appendix E.**

5.2 Video Alignment

Training Setting. Our pretrained model p_{ref} is an internal, research-purpose video generation model based on Transformer architecture [53], which is trained using rectified flow (see Appendix 7 for details). Following SD3 [15], all alignment experiments fine-tune the Transformer using LoRA [24]. For training-based alignment methods, including SFT, Flow-DPO, and Flow-RWR, we adopt VideoReward to provide the reward signals. For reward guidance, we employ the *latent* reward model to generate rewards. For supervised fine-tuning (SFT), we utilize only the “chosen data”. Following Rafailov et al. [61], we employ VideoReward as the ground-truth reward model to simulate human feedback and relabel our training dataset, ensuring that models optimised on these synthetic labels can be evaluated fairly by the same reward function. Hyperparameter settings are provided in the Appendix 1.

Evaluation. We evaluate performance using both automatic and human assessments. For automatic evaluation, we measure the *win rate* (via VideoReward) and the *Vbench* score. **The win rate** is the proportion of cases where VideoReward assigns a higher reward to the aligned model than to the pretrained model. **Vbench** is a fine-grained T2V benchmark that assesses Quality and Semantic alignment. For human evaluation, each sample is reviewed by two annotators, with a third annotator resolving disagreements. Identical random seeds are used across methods for strict comparability. Our evaluation uses prompts from *Vbench*, *VideoGen-Eval*, and a new *TA-Hard* set that stresses



Figure 4: Comparison of videos generated by the original model and the Flow-DPO aligned model.

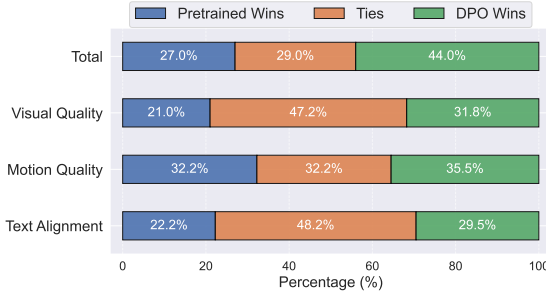


Figure 5: Human evaluation of Flow-DPO aligned model vs. pretrained model on VideoGen-Eval.

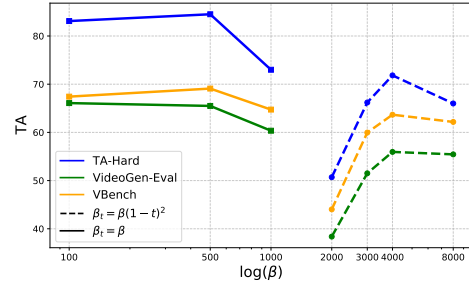


Figure 6: Accuracy of time-dependent β_t v.s. constant β for TA.

complex semantics, since we notice that the VBench and VideoGen-Eval prompts are relatively straightforward in terms of text alignment. Appendix L lists a subset of TA-Hard prompts.

Multi-dimensional Alignment. We use linear scalarization [40], $r = \frac{1}{3}(r_{vq} + r_{mq} + r_{ta})$, to aggregate multi-dimensional preferences into a single score, and forming a dataset $D = \{(y, x^w, x^l)\}$ where $r(x^w, y) > r(x^l, y)$. Table 3 shows that Flow-DPO with a constant β significantly improves over the pretrained model across all three dimensions and outperforms both SFT and Flow-RWR. In contrast, Flow-DPO with timestep-dependent β underperforms the pretrained model on TA, despite high VQ/MQ reward win rates due to reward hacking. Meanwhile, the constant- β variant achieves high VQ/MQ scores without such hacking issues, suggesting it learns TA more effectively. We discuss this further in Sec. 5.2. Figures 5 confirms these findings in human studies.

Single-dimensional Alignment. We also investigate the ability of different methods on a specific task: TA. Tab. 4 shows that Flow-DPO with a constant β achieves best performance across all datasets.

Reward Guidance. We apply linear scalarization to combine dimension rewards into a weighted sum and backpropagate gradients to the noised latent with guidance strength $w = 100$. Table 5 shows that users can steer generation toward custom trade-offs by simply choosing custom weights.

Table 5: Reward guidance on VideoGen-Eval.

VQ:MQ:TA	VQ	MQ	TA
0:0:1	60.56	46.48	70.42
0.1:0.1:0.8	66.50	63.73	60.86
0.1:0.1:0.6	68.94	67.59	53.28
0.5:0.5:0	86.43	93.23	26.65

Ablation on β . We meticulously adapted diffusion-DPO to flow-based models, resulting in Equation 6, where $\beta_t = \beta(1-t)^2$. Figure 6 shows that under various β values, a constant β consistently outperforms the timestep-dependent variant. This is likely because varying β across timesteps leads to uneven training [18], given that T2V models use shared weights for different noise levels.

6 Conclusion

We present a large-scale preference dataset of 182k human annotations covering visual quality, motion quality, and text alignment for modern video generation models. Building on this dataset, we introduce *VideoReward*, a multi-dimensional video reward model, and establish the *VideoGen-RewardBench* benchmark for more accurate and fair evaluations. From a unified reinforcement learning perspective, we further propose three alignment algorithms tailored to flow-based video generation, demonstrating their effectiveness in practice.

7 Limitations & Future Work

In our experiments, excessive training with Flow-DPO led to a significant deterioration in model quality, despite improvements in alignment across specific dimensions. To prevent this decline, we employed LoRA training. Future work can explore the simultaneous use of high-quality data for supervised learning during DPO training, aiming to preserve video quality while enhancing alignment. Additionally, our algorithms have been validated on text-to-video tasks; future work can extend the validation to other conditional generation tasks, such as image-to-video generation. Moreover, despite our efforts like incorporating tie annotations into the modeling, our VideoReward model still carries a potential risk of reward hacking, where human assessments indicate a marked decrease in video quality, yet the reward model continues to assign high scores. This issue arises because the reward function is differentiable, making it susceptible to manipulation. Future research should focus on developing more robust reward models, potentially by incorporating uncertainty estimates and increasing data augmentation. Additionally, there is potential to apply more RLHF algorithms, such as PPO, to flow-based video generation tasks.

Acknowledgements

This work was supported by the JC STEM Lab of AI for Science and Engineering, which is funded by The Hong Kong Jockey Club Charities Trust and the Research Grants Council of Hong Kong (Project No. CUHK14213224). It was also supported by the National Natural Science Foundation of China (Grant No. 62576191).

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *arXiv preprint arXiv:2305.13301*, 2023.
- [3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.
- [4] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- [5] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024.
- [6] Capcut. Dreamina. <https://dreamina.capcut.com/ai-tool/home>, 2024.
- [7] Chaofeng Chen, Annan Wang, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Enhancing diffusion models with text-encoder reinforcement learning. In *European Conference on Computer Vision*, pages 182–198. Springer, 2024.
- [8] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In

- Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7310–7320, 2024.
- [9] Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *arXiv preprint arXiv:2309.17400*, 2023.
 - [10] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39 (1):1–22, 1977.
 - [11] Daniel Deutsch, George Foster, and Markus Freitag. Ties matter: Meta-evaluating modern metrics with pairwise accuracy and tie calibration. *arXiv preprint arXiv:2305.14324*, 2023.
 - [12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
 - [13] Carles Domingo-Enrich, Michal Drozdal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. *arXiv preprint arXiv:2409.08861*, 2024.
 - [14] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.
 - [15] 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.
 - [16] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.
 - [17] Hiroki Furuta, Heiga Zen, Dale Schuurmans, Aleksandra Faust, Yutaka Matsuo, Percy Liang, and Sherry Yang. Improving dynamic object interactions in text-to-video generation with ai feedback. *arXiv preprint arXiv:2412.02617*, 2024.
 - [18] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo. Efficient diffusion training via min-snr weighting strategy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7441–7451, 2023.
 - [19] Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhramil Chandra, Ziyang Jiang, Aaran Arulraj, et al. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. *arXiv preprint arXiv:2406.15252*, 2024.
 - [20] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
 - [21] HidreamAI. Hidreamai. <https://www.hidreamai.com/>, 2024.
 - [22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
 - [23] Wenyi Hong, Weihang Wang, Ming Ding, Wenmeng Yu, Qingsong Lv, Yan Wang, Yean Cheng, Shiyu Huang, Junhui Ji, Zhao Xue, et al. Cogvlm2: Visual language models for image and video understanding. *arXiv preprint arXiv:2408.16500*, 2024.
 - [24] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

- [25] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023.
- [26] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.
- [27] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- [28] Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhui Chen. Mantis: Interleaved multi-image instruction tuning. *arXiv preprint arXiv:2405.01483*, 2024.
- [29] Dongfu Jiang, Max Ku, Tianle Li, Yuansheng Ni, Shizhuo Sun, Rongqi Fan, and Wenhui Chen. Genai arena: An open evaluation platform for generative models. *arXiv preprint arXiv:2406.04485*, 2024.
- [30] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [31] 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.
- [32] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.
- [33] Kuaishou. Kling ai. <https://klingai.kuaishou.com/>, 2024.
- [34] Pika Labs. Pika 1.0. <https://pika.art/>, 2023.
- [35] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024.
- [36] Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *arXiv preprint arXiv:2302.12192*, 2023.
- [37] Jiachen Li, Weixi Feng, Tsu-Jui Fu, Xinyi Wang, Sugato Basu, Wenhui Chen, and William Yang Wang. T2v-turbo: Breaking the quality bottleneck of video consistency model with mixed reward feedback. *arXiv preprint arXiv:2405.18750*, 2024.
- [38] Jiachen Li, Qian Long, Jian Zheng, Xiaofeng Gao, Robinson Piramuthu, Wenhui Chen, and William Yang Wang. T2v-turbo-v2: Enhancing video generation model post-training through data, reward, and conditional guidance design. *arXiv preprint arXiv:2410.05677*, 2024.
- [39] Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*, 2023.
- [40] Kaiwen Li, Tao Zhang, and Rui Wang. Deep reinforcement learning for multiobjective optimization. *IEEE transactions on cybernetics*, 51(6):3103–3114, 2020.
- [41] Youwei Liang, Junfeng He, Gang Li, Peizhao Li, Arseniy Klimovskiy, Nicholas Carolan, Jiao Sun, Jordi Pont-Tuset, Sarah Young, Feng Yang, et al. Rich human feedback for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19401–19411, 2024.
- [42] 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*, 2024.

- [43] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699, 2024.
- [44] Zicheng Lin, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. Criticbench: Benchmarking llms for critique-correct reasoning. *arXiv preprint arXiv:2402.14809*, 2024.
- [45] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.
- [46] Jinsong Liu, Dongdong Ge, and Ruihao Zhu. Reward learning from preference with ties. *arXiv preprint arXiv:2410.05328*, 2024.
- [47] Runtao Liu, Haoyu Wu, Zheng Ziqiang, Chen Wei, Yingqing He, Renjie Pi, and Qifeng Chen. Videodpo: Omni-preference alignment for video diffusion generation. *arXiv preprint arXiv:2412.14167*, 2024.
- [48] 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.
- [49] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tiejong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22139–22149, 2024.
- [50] LumaLabs. Dream machine. <https://lumalabs.ai/dream-machine>, 2024.
- [51] OpenAI. Video generation models as world simulators. <https://openai.com/index/video-generation-models-as-world-simulators>, 2024.
- [52] 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.
- [53] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.
- [54] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.
- [55] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pages 745–750, 2007.
- [56] PixVerse. Pixverse. <https://pixverse.ai/>, 2024.
- [57] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.
- [58] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023.
- [59] Mihir Prabhudesai, Russell Mendonca, Zheyang Qin, Katerina Fragkiadaki, and Deepak Pathak. Video diffusion alignment via reward gradients. *arXiv preprint arXiv:2407.08737*, 2024.
- [60] 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.

- [61] 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, 2024.
- [62] PV Rao and Lawrence L Kupper. Ties in paired-comparison experiments: A generalization of the bradley-terry model. *Journal of the American Statistical Association*, 62(317):194–204, 1967.
- [63] Runway. Gen-2: Generate novel videos with text, images or video clips. <https://runwayml.com/research/gen-2>, 2023.
- [64] Runway. Gen-3. <https://runwayml.com/>, 2024.
- [65] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [66] Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pages 32483–32498. PMLR, 2023.
- [67] 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.
- [68] Zhiwei Tang, Jiangweizhi Peng, Jiasheng Tang, Mingyi Hong, Fan Wang, and Tsung-Hui Chang. Tuning-free alignment of diffusion models with direct noise optimization. *arXiv preprint arXiv:2405.18881*, 2024.
- [69] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [70] VegaAI. Vegaai. <https://www.vegaai.net/>, 2023.
- [71] 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.
- [72] 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.
- [73] Yibin Wang, Zhiyu Tan, Junyan Wang, Xiaomeng Yang, Cheng Jin, and Hao Li. Lift: Leveraging human feedback for text-to-video model alignment. *arXiv preprint arXiv:2412.04814*, 2024.
- [74] Tianhe Wu, Jian Zou, Jie Liang, Lei Zhang, and Kede Ma. Visualquality-r1: Reasoning-induced image quality assessment via reinforcement learning to rank. *arXiv preprint arXiv:2505.14460*, 2025.
- [75] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.
- [76] 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.
- [77] 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, 2024.

- [78] Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihang Shen, Xiaolong Zhu, and Xiu Li. 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*, pages 8941–8951, 2024.
- [79] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.
- [80] Po-Hung Yeh, Kuang-Huei Lee, and Jun-Cheng Chen. Training-free diffusion model alignment with sampling demons. *arXiv preprint arXiv:2410.05760*, 2024.
- [81] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23174–23184, 2023.
- [82] Hangjie Yuan, Shiwei Zhang, Xiang Wang, Yujie Wei, Tao Feng, Yining Pan, Yingya Zhang, Ziwei Liu, Samuel Albanie, and Dong Ni. Instructvideo: instructing video diffusion models with human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6463–6474, 2024.
- [83] Huizhuo Yuan, Zixiang Chen, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning of diffusion models for text-to-image generation. *arXiv preprint arXiv:2402.10210*, 2024.
- [84] Ailing Zeng, Yuhang Yang, Weidong Chen, and Wei Liu. The dawn of video generation: Preliminary explorations with sora-like models. *arXiv preprint arXiv:2410.05227*, 2024.
- [85] 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.
- [86] Sixian Zhang, Bohan Wang, Junqiang Wu, Yan Li, Tingting Gao, Di Zhang, and Zhongyuan Wang. Learning multi-dimensional human preference for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8018–8027, 2024.
- [87] Yinan Zhang, Eric Tzeng, Yilun Du, and Dmitry Kislyuk. Large-scale reinforcement learning for diffusion models. In *European Conference on Computer Vision*, pages 1–17. Springer, 2024.
- [88] Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky TQ Chen. Guided flows for generative modeling and decision making. *arXiv preprint arXiv:2311.13443*, 2023.
- [89] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.
- [90] Zhanhui Zhou, Jie Liu, Chao Yang, Jing Shao, Yu Liu, Xiangyu Yue, Wanli Ouyang, and Yu Qiao. Beyond one-preference-for-all: Multi-objective direct preference optimization. *arXiv preprint arXiv:2310.03708*, 2023.

Appendix of Improving Video Generation with Human Feedback

Our Appendix consists of 8 sections. Readers can click on each section number to navigate to the corresponding section:

- Section [A](#) provides a review of related works.
- Section [B](#) provides detailed derivations and lemma proofs for Flow-DPO, Flow-RWR, and Flow-NRG.
- Section [C](#) provides pseudo-code of Flow-DPO and Flow-NRG.
- Section [D](#) presents the architecture of the internal video diffusion model employed in our work.
- Section [E](#) offers ablation study on key design choices of our reward model.
- Section [F](#) summarizes dataset statistics.
- Section [G](#) explains details of human annotation and guidelines.
- Section [H](#) provides details about reward model evaluation, including a comparison of the two evaluation benchmarks, evaluation procedures of all methods, and the metrics employed.
- Section [I](#) lists hyperparameters for our reward modeling and alignment algorithms.
- Section [J](#) provides additional visual comparisons between the original model and the Flow-DPO aligned model.
- Section [K](#) provides the input template used for reward model.
- Section [L](#) provides part of prompts used in our TA-Hard dataset.
- Section [M](#) provides broader impacts.

A Related Works

Evaluation and Reward Models. Evaluation models and reward models play a pivotal role in aligning generative models with human preferences. Earlier approaches and benchmarks [25, 49, 26] relied on metrics like FID [20] and CLIP scores [60] to assess visual quality and semantic consistency. Recent works [75, 77, 31, 86, 41] have shifted towards utilizing human preference datasets to train CLIP-based models, enabling them to predict preference scores with improved accuracy. With the advent of large vision-language models (VLMs) [1, 72], their powerful capabilities in visual understanding and text-visual alignment make them a natural proxy for reward modeling. A common approach involves replacing the token classification head of VLMs with a regression head that predicts multi-dimensional scores for diverse evaluation tasks.

Two main learning paradigms have emerged based on the type of human annotation. The first paradigm relies on point-wise regression, where the model learns to fit annotated scores [19] or labels [76] directly. Another paradigm focuses on pair-wise comparisons [74], leveraging Bradley-Terry (BT) [4] loss or rank loss to model relative preferences, which is largely unexplored for video reward model. Beyond these methods, some works [73] also leverage the intrinsic reasoning capabilities of VLMs through VLM-as-a-judge [39, 44], where VLMs are adopted to generate preference judgments or scores in textual format through instruction tuning. Despite these promising advances, most existing video reward models primarily focus on legacy video generation models, typically from the pre-Sora [51] era, which are constrained by short video durations and relatively low quality. Furthermore, the technical choices underlying the vision reward models remain underexplored. Our work seeks to address these limitations by focusing more on modern video generation models and investigating a broader range of reward modeling strategies.

Alignment for Image & Video Generation. In large language models, Reinforcement Learning from Human Feedback (RLHF) improves alignment with human preferences, enhancing response quality [52, 27]. Similar methods have been applied to image generation, using reward models or human preference data to align pretrained models. Key approaches include: (1) direct backpropagation with reward signals [58, 9, 77, 59]; (2) Reward-Weighted Regression (RWR) [54, 36, 17]; (3) DPO-based policy optimization [61, 71, 42, 14, 78, 42, 83, 47, 85, 17]; (4) PPO-based policy gradients [65] [2, 16, 87, 7]; and (5) training-free alignment [80, 68, 66]. These methods have successfully aligned image generation models with human preferences, improving aesthetics and semantic consistency. They focus on improving the accuracy of rewards on noised images using reward models trained on clean images, whereas our Flow-NRG directly trains noise-aware reward models to obtain accurate gradients. Our work applies the DPO algorithm [61, 71] to flow matching in video generation. Concurrent work [13] also explores similar things in image generation. However, they reports worse performance for the DPO-aligned model compared to the unaligned one. We argue that the originally derived Flow-DPO algorithm imposes a stronger KL constraint at lower noise steps, resulting in suboptimal performance. In contrast, using a constant KL constraint significantly improves performance on certain tasks. Some prior work explores aligning video generation models using direct backpropagation with differentiable rewards [82, 38, 37, 59], often relying on image reward models [31, 75]. However, these approaches cannot be directly applied to modern T2V generation with a VLM-based video reward and large VAE decoders, as they exceed the memory limits of existing GPU setups, requiring specialized engineering techniques to handle.

Discussion with Concurrent Video Alignment Works. Several concurrent works also explore aligning video generation models using feedback. Furuta et al. [17] derives a unified probabilistic objective for offline RL fine-tuning of text-to-video models for DPO and RWR. VideoDPO [47] introduces a re-weighting factor on the Diffusion-DPO loss to adjust the impact of each pair, encouraging the model to learn more effectively from pairs with clearer distinctions. VisionReward [76] ensures that all dimensions of the chosen data outperform those of the rejected data, addressing the issue of confounding factors in preference data. OnlineVPO [85] presents an online DPO algorithm to tackle off-policy optimization. LIFT [73] proposes applying RWR on synthesized datasets while simultaneously performing supervised learning on real video-text datasets, as synthesized videos often suffer from low temporal consistency. All of these works use Diffusion-DPO or Diffusion-RWR and focus on aligning diffusion-based video generation models, where the videos in the preference datasets are either generated by earlier open-source models or use image reward models directly. In contrast, our work explores alignment techniques for advanced flow-based video generation. We extend Diffusion-DPO into Flow-DPO, but our derivation reveals that the parameter β (the KL divergence constraint) in Flow-DPO is timestep-dependent, which leads to suboptimal performance on certain tasks. However, fixing β resolves this issue. SPO [42] also assumes that the preference order between pairs of images remains consistent when adding the same noise, and constructs win-lose pairs for noised images, proposing step-aware preference optimization. Our work differs from SPO in that while SPO focuses on improving the training method DPO, our Flow-NRG specifically targets training noise-aware reward model on noised videos.

B Details of the Derivation

B.1 Relation between Velocity and Noise

Lemma B.1. *Let $X_0 \sim q$ be a real data sample drawn from the true data distribution and $X_1 \sim p$ be a noise sample, where $X_0, X_1 \in \mathbb{R}^d$. Define $\mathbf{v}_t(\mathbf{x}_t | X_0, X_1)$ to be the conditional velocity field specified by a Rectified Flow [48], and let $\mathbf{v}_t^{pred}(\mathbf{x}_t)$ be the predicted marginal velocity field. Then the L2 error of the noise prediction is related to the L2 error of the velocity field prediction by*

$$\|X_1 - X_1^{pred}(\mathbf{x}_t, t)\|^2 = (1 - t)^2 \|\mathbf{v}_t(\mathbf{x}_t | X_0, X_1) - \mathbf{v}_t^{pred}(\mathbf{x}_t)\|^2. \quad (12)$$

Proof. The Rectified Flow is a time-dependent flow $\psi : [0, 1] \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ for $t \in [0, 1]$, defined by

$$\psi(X_0, X_1) = (1 - t) X_0 + t X_1.$$

By definition, the *marginal* velocity field $\mathbf{v}_t(\mathbf{x}_t)$ is

$$\begin{aligned}
\mathbf{v}_t(\mathbf{x}_t) &= \mathbb{E}[\mathbf{v}_t(X_t | X_0, X_1) \mid X_t = \mathbf{x}_t] \\
&= \mathbb{E}[\dot{\psi}(X_t | X_0, X_1) \mid X_t = \mathbf{x}_t] \\
&= \mathbb{E}[X_1 - X_0 \mid X_t = \mathbf{x}_t] \\
&= \mathbb{E}\left[\frac{X_1 - ((1-t)X_0 + tX_1)}{1-t} \mid X_t = \mathbf{x}_t\right] \\
&= \mathbb{E}\left[\frac{X_1 - \mathbf{x}_t}{1-t} \mid X_t = \mathbf{x}_t\right] \\
&= \frac{\mathbb{E}[X_1 \mid X_t = \mathbf{x}_t] - \mathbf{x}_t}{1-t}.
\end{aligned} \tag{13}$$

Meanwhile, the *conditional* velocity field $\mathbf{v}_t(\mathbf{x}_t | X_0, X_1)$ is given by

$$\mathbf{v}_t(\mathbf{x}_t | X_0, X_1) = \frac{X_1 - \mathbf{x}_t}{1-t}. \tag{14}$$

Substituting equation 14 into equation 13, we obtain

$$\|\mathbf{v}_t(\mathbf{x}_t | X_0, X_1) - \mathbf{v}_t(\mathbf{x}_t)\|^2 = \frac{\|X_1 - \mathbb{E}[X_1 \mid X_t = \mathbf{x}_t]\|^2}{(1-t)^2}.$$

Assuming that

$$X_1^{\text{pred}}(\mathbf{x}_t, t) = \mathbb{E}[X_1 \mid X_t = \mathbf{x}_t] \quad \text{and} \quad \mathbf{v}_t^{\text{pred}}(\mathbf{x}_t) = \mathbf{v}_t(\mathbf{x}_t).$$

Consequently,

$$\|X_1 - X_1^{\text{pred}}(\mathbf{x}_t, t)\|^2 = (1-t)^2 \|\mathbf{v}_t(\mathbf{x}_t | X_0, X_1) - \mathbf{v}_t^{\text{pred}}(\mathbf{x}_t)\|^2.$$

□

B.2 Reward as Classifier Guidance

We begin by citing a lemma from the Guided Flows paper [88].

Lemma B.2. *Let $p_t(\mathbf{x}|\mathbf{y})$ be a Gaussian Path defined by a scheduler (α_t, σ_t) , i.e., $p_t(\mathbf{x}|x_0) = \mathcal{N}(\mathbf{x}|\alpha_t x_0, \sigma_t^2 I)$ where $\mathbf{y} \in \mathbb{R}^k$ is a conditioning variable, then its generating velocity field $\mathbf{v}_t(\mathbf{x}|\mathbf{y})$ is related to the score function $\nabla \log p_t(\mathbf{x}|\mathbf{y})$ by*

$$\mathbf{v}_t(\mathbf{x}|\mathbf{y}) = a_t \mathbf{x} + b_t \nabla \log p_t(\mathbf{x}|\mathbf{y}), \tag{15}$$

where

$$a_t = \frac{\dot{\alpha}_t}{\alpha_t}, \quad b_t = (\dot{\alpha}_t \sigma_t - \alpha_t \dot{\sigma}_t) \frac{\sigma_t}{\alpha_t}. \tag{16}$$

See Appendix A.61 of the Guided Flows paper [88] for detailed derivations.

If we define

$$\begin{aligned}
\tilde{v}_t(\mathbf{x}_t|\mathbf{y}) &= \mathbf{v}_t(\mathbf{x}_t|\mathbf{y}) + w[a_t \mathbf{x}_t + b_t \nabla r(\mathbf{x}_t, \mathbf{y}) - \mathbf{v}_t(\mathbf{x}_t|\mathbf{y})] \\
&= \mathbf{v}_t(\mathbf{x}_t|\mathbf{y}) + w[a_t \mathbf{x}_t + b_t \nabla \log \exp(r(\mathbf{x}_t, \mathbf{y})) - \mathbf{v}_t(\mathbf{x}_t|\mathbf{y})] \\
&= (1-w)\mathbf{v}_t(\mathbf{x}_t|\mathbf{y}) + w[a_t \mathbf{x}_t + b_t \nabla \log \exp(r(\mathbf{x}_t, \mathbf{y}))] \\
&= a_t \mathbf{x}_t + b_t \nabla [(1-w) \log p_t(\mathbf{x}_t|\mathbf{y}) + w \log \exp(r(\mathbf{x}_t, \mathbf{y}))] \\
&= a_t \mathbf{x}_t + b_t \nabla \log \tilde{p}_t(\mathbf{x}_t|\mathbf{y})
\end{aligned} \tag{17}$$

where $\tilde{p}_t(\mathbf{x}_t|\mathbf{y}) \propto p_t(\mathbf{x}_t|\mathbf{y})^{1-w} [\exp(r(\mathbf{x}_t, \mathbf{y}))]^w$. We change our goal from sampling from the distribution $p_t(\mathbf{x}_t|\mathbf{y})$ to sampling from the distribution $\tilde{p}_t(\mathbf{x}_t|\mathbf{y})$.

We note that this analysis shows that Reward Guided Flows are guaranteed to sample from $\tilde{q}(\cdot|\mathbf{y})$ at time $t = 1$ if the probability path $\tilde{p}_t(\cdot|\mathbf{y})$ is close to the marginal probability path $\int p_t(\cdot|x_1) \tilde{q}(x_1|\mathbf{y}) dx_1$, but it is not clear to what extent this assumption holds in practice. This also means that $\tilde{p}_t(\mathbf{x}_t|\mathbf{y})$ is also a marginal gaussian path defined by $p_t(\mathbf{x}|\mathbf{x}_1) = \mathcal{N}(\mathbf{x}|\alpha_t \mathbf{x}_1, \sigma_t^2 I)$.

Simlirly, if we define

$$\begin{aligned}
\tilde{v}_t(\mathbf{x}_t|\mathbf{y}) &= \mathbf{v}_t(\mathbf{x}_t|\mathbf{y}) + wb_t\nabla r(\mathbf{x}_t, \mathbf{y}) \\
&= \mathbf{v}_t(\mathbf{x}_t|\mathbf{y}) + wb_t\nabla \log \exp(r(\mathbf{x}_t, \mathbf{y})) \\
&= a_t\mathbf{x}_t + b_t\nabla [\log p_t(\mathbf{x}_t|\mathbf{y}) + w \log \exp(r(\mathbf{x}_t, \mathbf{y}))] \\
&= a_t\mathbf{x}_t + b_t\nabla \log \tilde{p}_t(\mathbf{x}_t|\mathbf{y})
\end{aligned} \tag{18}$$

where $\tilde{p}_t(\mathbf{x}_t|\mathbf{y}) \propto p_t(\mathbf{x}_t|\mathbf{y})[\exp(r(\mathbf{x}_t, \mathbf{y}))]^w$. We change our goal from sampling from the distribution $p_t(\mathbf{x}_t|\mathbf{y})$ to sampling from the distribution $\tilde{p}_t(\mathbf{x}_t|\mathbf{y})$.

Reward Guidance for Rectified Flow. Rectified Flow [48] is also a Gaussian path defined by

$$\mathbf{x}_t = (1 - t)\mathbf{x}_0 + t\mathbf{x}_1 \tag{19}$$

where $\mathbf{x}_1$ is from normal Gaussian distribution. Then

$$p_t(\mathbf{x} | \mathbf{x}_0) = \mathcal{N}(\mathbf{x} | (1 - t)\mathbf{x}_0, t^2 I)$$

where $\alpha_t = 1 - t, \sigma_t = t$. Then we get

$$a_t = \frac{1}{t - 1}, b_t = \frac{t}{t - 1}. \tag{20}$$

Eq. 17 becomes

$$\tilde{v}_t(\mathbf{x}_t|\mathbf{y}) = \mathbf{v}_t(\mathbf{x}_t|\mathbf{y}) + w[\frac{1}{1-t}\mathbf{x}_t + \frac{t}{1-t}\nabla r(\mathbf{x}_t, \mathbf{y}) + \mathbf{v}_t(\mathbf{x}_t|\mathbf{y})]. \tag{21}$$

Eq. 18 becomes

$$\tilde{v}_t(\mathbf{x}_t|\mathbf{y}) = \mathbf{v}_t(\mathbf{x}_t|\mathbf{y}) - w\frac{t}{1-t}\nabla r(\mathbf{x}_t, \mathbf{y}). \tag{22}$$

We use Eq. 17 or Eq. 18 to guide inference through reward model.

C Pseudo-code of Flow-DPO and Flow-NRG

The Pytorch-style implementation of the Flow-DPO loss (Eq. 6) is shown below:

```
def loss(model, ref_model, x_w, x_l, c, beta):
    """
    # model: Flow model that takes prompt condition c and timestep as inputs and
    #           predicts velocity
    # ref_model: Frozen initialization of the model
    # x_w: Preferred Image (latents in this work)
    # x_l: Non-Preferred Image (latents in this work)
    # c: Conditioning (text in this work)
    # beta: Regularization Parameter
    """
    timestep = torch.rand(len(x_w))
    noise = torch.randn_like(x_w)
    noisy_x_w = (1 - timestep) * x_w + timestep * noise
    noisy_x_l = (1 - timestep) * x_l + timestep * noise
    velocity_w_pred = model(noisy_x_w, c, timestep)
    velocity_l_pred = model(noisy_x_l, c, timestep)
    velocity_ref_w_pred = ref_model(noisy_x_w, c, timestep)
    velocity_ref_l_pred = ref_model(noisy_x_l, c, timestep)
    velocity_w = noise - x_w
    velocity_l = noise - x_l

    model_w_err = (velocity_w_pred - velocity_w).norm().pow(2)
    model_l_err = (velocity_l_pred - velocity_l).norm().pow(2)
    ref_w_err = (velocity_ref_w_pred - velocity_w).norm().pow(2)
    ref_l_err = (velocity_ref_l_pred - velocity_l).norm().pow(2)
    w_diff = model_w_err - ref_w_err
    l_diff = model_l_err - ref_l_err
    inside_term = -0.5 * beta * (w_diff - l_diff)
    loss = -1 * log(sigmoid(inside_term))
    return loss
```

The Pytorch-style implementation of the reward guidance (Eq. 11) is shown below:

```
def reward_guidance(model, reward_model, prompt_embeds, latents, timesteps,
                    reward_weight, rg_scale, cfg_scale):
    """
    # model: Flow model that predicts velocity given latents and conditions
    # reward_model: Model that evaluates the quality of latents based on prompt
    #           embeddings and timestep
    # prompt_embeds: Embeddings of the text prompts
    # latents: Initial noise
    # timesteps: Sequence of timesteps
    # reward_weight: weighting coefficient of multi-dimensional rewards
    # rg_scale: scale factor for reward guidance
    # cfg_scale: scale factor for classifier free guidance
    """
    dts = timesteps[:-1] - timesteps[1:]
    for i, t in enumerate(timesteps):
        v_pred = model(latents, prompt_embeds, t)
        if cfg_scale != 1.0:
            v_pred_uncond = model(latents, None, t) # unconditional prediction
            v_pred = v_pred_uncond + cfg_scale * (v_pred - v_pred_uncond)
        latents = latents.detach().requires_grad_(True)
        reward = reward_model(latents, prompt_embeds, t)
        reward = (reward * reward_weight).sum()
        reward_guidance = torch.autograd.grad(reward, latents)
        if t != 1:
            v_pred = v_pred - rg_scale * t / (1 - t) * reward_guidance
        latents = latents - dts[i] * v_pred
    return latents
```

D Architecture of Internal Video Diffusion Model

Our work employs an internal text-to-video foundation model, which is a Transformer-based latent diffusion model [53] with 1B parameters. The model integrates a 3D VAE to encode video data into a latent space, alongside a Transformer-based video diffusion model. Each Transformer block is instantiated as a sequence of 2D spatial-attention, 3D self-attention, cross-attention, and FFN modules. Importantly, the model is trained under Flow Matching framework. An illustration of the model architecture is provided in Fig 7.

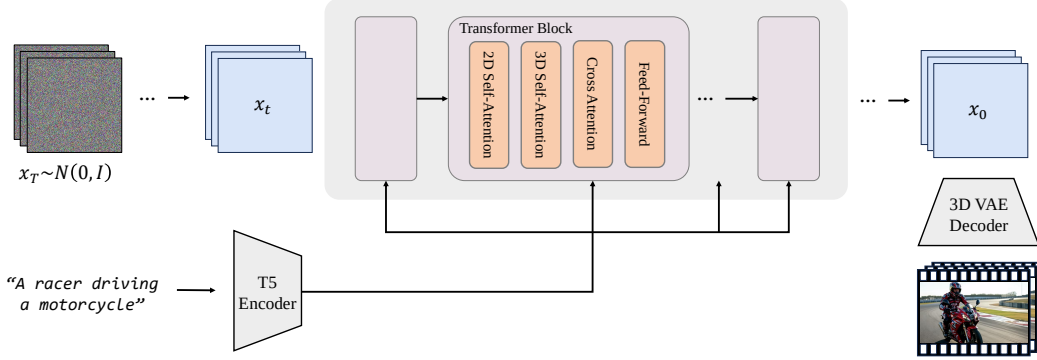


Figure 7: Architecture of Our Internal Video Latent Diffusion Model Backbone.

E Ablation Study of Our Reward Model

Ablation of VideoReward To further validate the key design choices of our reward model discussed in Sec.3.2, we conduct an ablation study on the evaluation benchmark, providing a quantitative supplement to our analysis in Tab. 6. We compare three reward model variants: *regression-style*, *Bradley-Terry*, and *Bradley-Terry-With-Tied*. The BT model slightly outperforms the regression-style model, likely due to the advantages of pairwise annotations, which better capture ranking relationships and are more robust to annotation noise. The BTT model matches the BT model on ties-excluded accuracy but significantly improves ties-included accuracy, as its explicit handling of tied pairs helps it learn a more robust decision boundary, capturing neutral relationships in ambiguous cases. Additionally, we find that using separate tokens for each reward attribute, instead of a shared last token further improves performance. This design better represents distinct reward aspects, enhancing alignment with human preferences.

Table 6: Ablation study on reward model type and seprate tokens. **Bold**: Best Performance.

Variants	RM Type	Separate Tokens	GenAI-Bench		VideoGen-RewardBench							
			Overall Accuracy		Overall Accuracy		VQ Accuracy		MQ Accuracy		TA Accuracy	
			w/ Ties	w/o Ties	w/ Ties	w/o Ties	w/ Ties	w/o Ties	w/ Ties	w/o Ties	w/ Ties	w/o Ties
I	Regression		48.28	71.13	58.39	70.16	54.23	73.61	61.16	65.56	52.60	70.95
II	BT		47.74	71.21	61.22	72.58	52.33	77.10	59.43	73.50	53.06	71.62
III	BTT		48.27	70.89	61.50	73.39	60.52	76.31	64.64	72.40	53.55	72.12
IV	BTT	✓	49.41	72.89	61.26	73.59	59.68	75.66	66.03	74.70	53.80	72.20

Ablation on Noisy-Latent Reward Training. We apply reward guidance using only MQ rewards on TA-Hard prompts. When trained with noised latents, the reward model successfully guides generation, yielding an MQ win rate of 74.6. By contrast, the model trained on clean latents fails to offer meaningful guidance for intermediate noised latents, achieving only a 38.6 win rate and underperforming even the unguided baseline. Following Yu et al. [81], we also try directly predicting x_0 from x_t and then backpropagating via $r(x_0, y)$. However, this method produces unusable videos without meaningful content.

F Statistics of Training Data

In Tab. 1 and Fig. 8, we provide a comprehensive statistics of our 182k-sized training dataset.

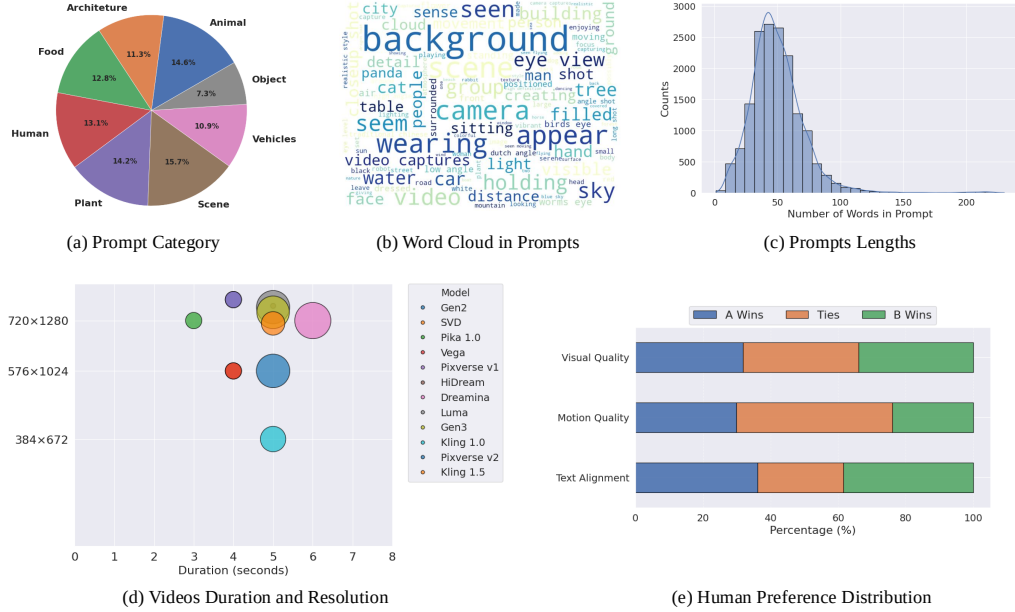


Figure 8: Statistics of our training data.

Previous reward models (e.g., VideoScore [19], LIFT [73]) were trained on datasets with mostly low resolutions ($\leq 480p$) and short durations (2s), whereas ours include recent T2V models with higher resolutions ($\geq 720p$) and longer clips (3–6s). The dataset statistics are shown in Tab. 7.

Table 7: Comparison of video preference datasets in terms of spatial resolution and temporal duration.

Dataset	Resolution	Duration
<i>Existing Preference Datasets</i>		
VideoFeedback [19]	$256 \times 256 - 576 \times 1024$	1s – 3s
VideoDPO [47]	320×512	2s
LiFT-HRA [73]	480×720	6s
<i>Our Datasets</i>		
Ours	$384 \times 672 - 768 \times 1408$	3s – 6s

G Details of Human Annotation

We provide additional details regarding the annotation process. First, annotators are provided with detailed scoring guidelines and undergo training sessions to ensure they fully understand the criteria; Tab 8 summarizes the key points for each dimension as outlined in the guidelines. Reference examples are provided to help annotators better grasp the evaluation standards. Each sample is evaluated by three independent annotators. For training and validation sets, annotators provide pairwise preference annotations and pointwise scores for Visual Quality (VQ), Motion Quality (MQ), and Temporal Alignment (TA). For VideoGen-RewardBench, annotators evaluate the same three aspects along with an additional Overall Quality, using only pairwise preferences. In cases where the annotators disagree on a sample, an additional reviewer is tasked with resolving the discrepancy. The final label is determined on the basis of the reviewer’s evaluation, ensuring consistency across the dataset. Furthermore, during the annotation process, all annotators are instructed to flag any content deemed unsafe. Videos identified as unsafe are excluded from the dataset, ensuring the safety of the data used for training and evaluation.

Table 8: Key points summary outlined in annotation guidelines for each evaluation dimension.

Evaluation Dimension	Key Points Summary
Visual Quality	Considering the following dimensions introduced by non-dynamic factors: - Image Reasonableness: The image should be objectively reasonable. - Clarity: The image should be clear and visually sharp. - Detail Richness: The level of intricacy in the generation of details. - Aesthetic Creativity: The generated videos should be aesthetically pleasing. - Safety: The generated video should not contain any disturbing or uncomfortable content.
Motion Quality	Considering the following dimensions in the dynamic process of the video: - Dynamic Stability: The continuity and stability between frames. - Dynamic Reasonableness: The dynamic movement should align with natural physical laws. - Motion Aesthetic Quality: The dynamic elements should be harmonious and not stiff. - Naturalness of Dynamic Fusion: The edges should be clear during the dynamic process. - Motion Clarity: The motion should be easy to identify. - Dynamic Degree: The movement should be clear, avoiding still scenes.
Text Alignment	Considering the relevance to the input text prompt description. - Subject Relevance: Relevance to the described subject characteristics and subject details. - Dynamic Information Relevance: Relevance to actions and postures as described in the text. - Environmental Relevance: Relevance of the environment to the input text. - Style Relevance: Relevance to the style descriptions, if exists. - Camera Movement Relevance: Relevance to the camera descriptions, if exists.

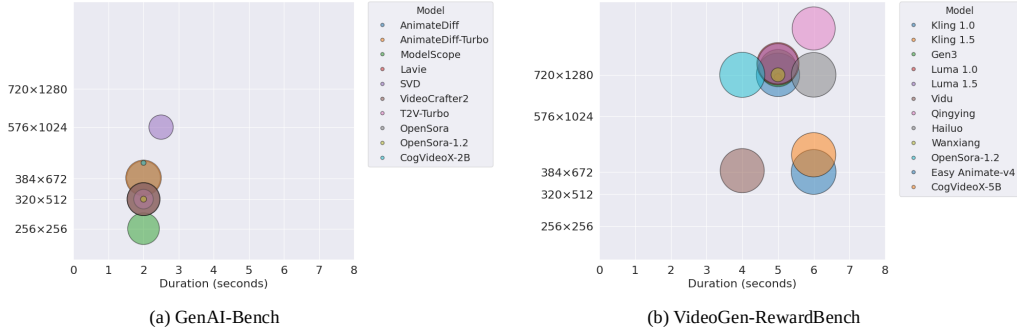


Figure 9: Video Duration and Resolution in GenAI-Bench and VideoGen-Reward Bench

H Details of Reward Model Evaluation

H.1 Evaluation Benchmarks

We evaluate our reward model using two benchmarks: GenAI-Bench [29] and VideoGen-RewardBench. **GenAI-Bench** is employed to assess the accuracy of the reward model in **evaluating pre-SOTA-era T2V models**, while **VideoGen-RewardBench** is used to evaluate its performance on **modern T2V models**. In this subsection, we describe both benchmarks, highlighting key parameters and differences in Fig 9 and Tab. 9. We also visualize the model coverage across the training sets of different baselines and the two evaluation benchmarks, as shown in the Fig 10.

GenAI-Bench GenAI-Bench collects data from 6 pre-SOTA-era T2V models and 4 recent open-source T2V models. Human annotations for overall quality are obtained through GenAI-Arena, resulting in a benchmark consisting of 10 T2V models, 508 prompts, and 1.9k pairs. As the videos

Table 9: Comparison between GenAI-Bench and VideoGen-RewardBench. Eariler Models indicates that pre-Sora-era T2V models, and Modern Models indicates that T2V models after Sora.

Benchmark	Prompts and Sampled Videos					Human Preference Annotations	
	#Samples	#Prompts	#Earlier Models	#Modern Models	#Duration	#Annotations	#Dimensions
GenAI-Bench	3784	508	7 (Open-Source)	3 (Open-Source)	2s - 2.5s	1891	1
VideoGen-RewardBench	4923	420	0	3 (Open-Source) 9 (Close-Source)	4s - 6s	26457	4

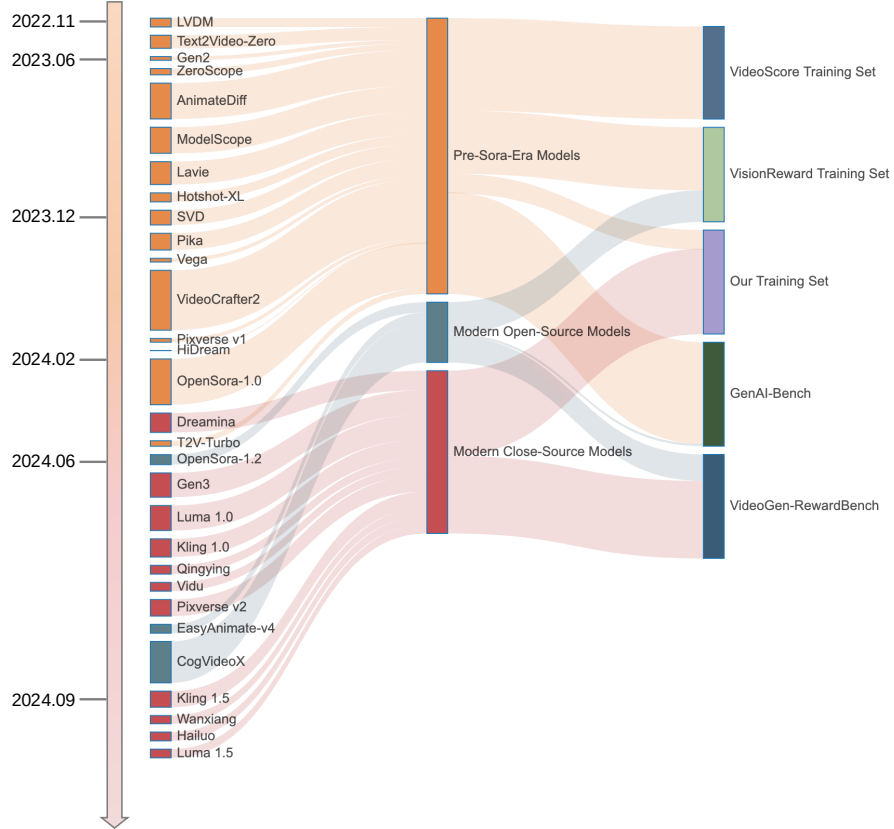


Figure 10: The model coverage across the training sets of different baselines and the two evaluation benchmarks. VideoScore, VisionReward, and GenAI-Bench primarily focus on pre-SoRA-era models, while our training set and VideoGen-RewardBench concentrate on state-of-the-art T2V models.

in GenAI-Bench predominantly originate from earlier video generation models, they typically have lower resolutions (most around 320x512) and shorter durations (2s-2.5s). We consider GenAI-Bench as a benchmark to assess the performance of reward models on early-generation T2V models.

VideoGen-RewardBench VideoGen-Eval [84] has open-sourced a dataset containing videos generated by 9 closed-source and 3 open-source models, designed to qualitatively visualize performance differences across models. Due to its high-quality data, broad coverage of the latest advancements in T2V models, and third-party nature, we leverage VideoGen-Eval to create a fair benchmark, VideoGen-RewardBench, for evaluating reward models’ performance on modern T2V models. We manually construct 26.5k video pairs and hire annotators to assess each pair’s Visual Quality, Motion Quality, Text Alignment, and Overall Quality, providing preference labels. Ultimately, VideoGen-RewardBench includes 12 T2V models, 420 prompts, and 26.5k pairs. This benchmark represents human preferences for state-of-the-art models, with videos featuring higher resolutions (480x720 - 576x1024), longer durations (4s - 6s), and improved quality. We use VideoGen-RewardBench as the primary benchmark to evaluate reward models’ performance on modern T2V models.

H.2 Comparison Methods

Random To eliminate the influence of metric calculations and benchmark distributions on our evaluation results, we introduce a special baseline: random scores. Specifically, for each triplet (*prompt*, *video A*, *video B*), we randomly sample r_A and r_B from a standard normal distribution, denoted as $r_A, r_B \sim \mathcal{N}(0, 1)$. We then calculate accuracy in the same manner as for the other models. The mathematical expectation of random scores for ties-excluded accuracy is $\mathbb{E}(acc) = \frac{1}{2}$, and the mathematical expectation of ties-included accuracy is $\mathbb{E}(acc) = \max(\frac{1}{3}, p(c = \text{"Ties"}))$.

VideoScore VideoScore [19] adopts Mantis-Idefics2-8B [28] as its base model and trains with point-wise data using MSE loss to model human preference scores. Since VideoScore predicts scores across multiple dimensions, and its dimension definitions differ from those in VideoGen-RewardBench, we compute both the overall accuracy and the dimension-specific (VQ, MQ, TA) accuracy by averaging the scores of five dimensions when conduct evaluation GenAI-Bench and VideoGen-RewardBench, consistent with the evaluation strategy outlined in their paper. The training data for VideoScore predominantly comes from pre-SOTA-era models, which explains its relatively better performance on GenAI-Bench, while accounts for the significant performance drop on VideoGen-RewardBench.

LiFT LiFT [73] adopts VILA-1.5-40B [43] as its base model and employs a VLM-as-a-judge approach. The reward model is trained through instruction tuning with inputs, preference scores along with a critic. The model generates video scores and reasons through next-token prediction. LiFT evaluates videos across three dimensions: Video Fidelity, Motion Smoothness, and Semantic Consistency, which are similar to the dimensions defined in VideoGen-RewardBench. We calculate the overall accuracy using the average scores of these three dimensions and compute the dimension-specific accuracy using the corresponding dimensional scores. LiFT predicts discrete scores on a 1-3 scale, which often leads to ties in pairwise comparisons. When calculating accuracy without ties, we randomly convert the predicted tie labels to chosen/rejected with a 50% probability, indicating that the model is unable to distinguish the relative quality between the two samples.

VisionReward VisionReward [76] adopts CogVLM2-Video-12B[23] as the base model and is trained to answer a set of judgment questions about the video with a binary "yes" or "no" response using cross-entropy loss. During inference, VisionReward evaluates 64 checklist items, providing converted into 1/0 scores. The final score is computed as the weighted average of these individual responses. We use the final score to calculate both the overall accuracy and the VQ/MQ/TA accuracy. VisionReward’s training data includes models from the pre-SOTA era models [8] as well as recent open-source T2V models [89, 79]. It performs well on GenAI-Bench and demonstrates reasonable capabilities on VideoGen-RewardBench.

Our Reward Model We adopts QWen2-VL-2B [72] as the base model and train it with pair-wise data using BTT loss in Eq. 2. Scores are normalized on the validation set and averaged to obtain overall scores for evaluation and optimization. When evaluating on VideoGen-RewardBench, we sample videos at 2 FPS and a resolution of 448×448, consistent with the training settings. We **calculate the overall accuracy by averaging the scores** across the three dimensions, and compute dimension-specific accuracies using the respective scores. For GenAI-Bench, we sample videos at 2 FPS and a resolution of 256×256, as the minimum resolution in GenAI-Bench is 256×256. Given the significant disparities in visual quality and motion between the GenAI-Bench videos and our training data, **we utilize only the predicted TA scores to calculate the overall score.**

H.3 Evaluation Metrics

Similarly to VisionReward [76], we report two accuracy metrics: ties-included accuracy [11] and ties-excluded accuracy. For ties-excluded accuracy, we exclude all data labeled as "ties" and use only data labeled as "A wins" or "B wins" for calculation. Since all competitors predict scores based on pointwise samples, we compute the rewards for each pair, convert the relative reward relationships into binary labels, and calculate classification accuracy. For ties-included accuracy, we adopt the tie calibration algorithm proposed in Algorithm 1 by Deutsch et al. [11]. This method traverses all possible tie thresholds, calculates three-class accuracy for each threshold, and selects the highest accuracy as the final metric.

I Hyperparameters

In all alignment experiments, we applied LoRA to fine-tune the transformer models’ linear layers, as our findings indicate that full parameter fine-tuning can degrade the model’s performance or potentially lead to model collapse.

Table 10: Hyperparameters for alignment algorithms

Algorithm-agnostic hyperparameters for SFT, Flow-RWR, Flow-DPO	
Training strategy	LoRA [24]
LoRA alpha	128
LoRA dropout	0.0
LoRA R	64
LoRA target-modules	q-proj,k-proj,v-proj,o-proj
Optimizer	Adam [30]
Learning rate	5e-6
Epochs	1
Batch size	64
GPUs	16 NVIDIA A800@80G
Flow-DPO	
β	500

Table 11: Hyperparameters for reward modeling.

VLM	
Training strategy	Full training for vision encoder LoRA for language model
LoRA alpha	128
LoRA dropout	0.0
LoRA R	64
LoRA target-modules	Linear layers in language model
Optimizer	Adam [30]
Learning rate	2e-6
Epochs	2
Batch size	32
GPUs	8 NVIDIA A800@80G
θ in Eq. 2	5.0
VDM	
Training strategy	Full training
Optimizer	Adam [30]
Learning rate	5e-6
Epochs	2
Batch size	144
Reward Dimension	3
GPUs	8 NVIDIA A800@80G

J Additional Qualitative Results

We present additional qualitative results generated by both the original model and the Flow-DPO aligned model, as shown in Fig.11 and Fig.12.

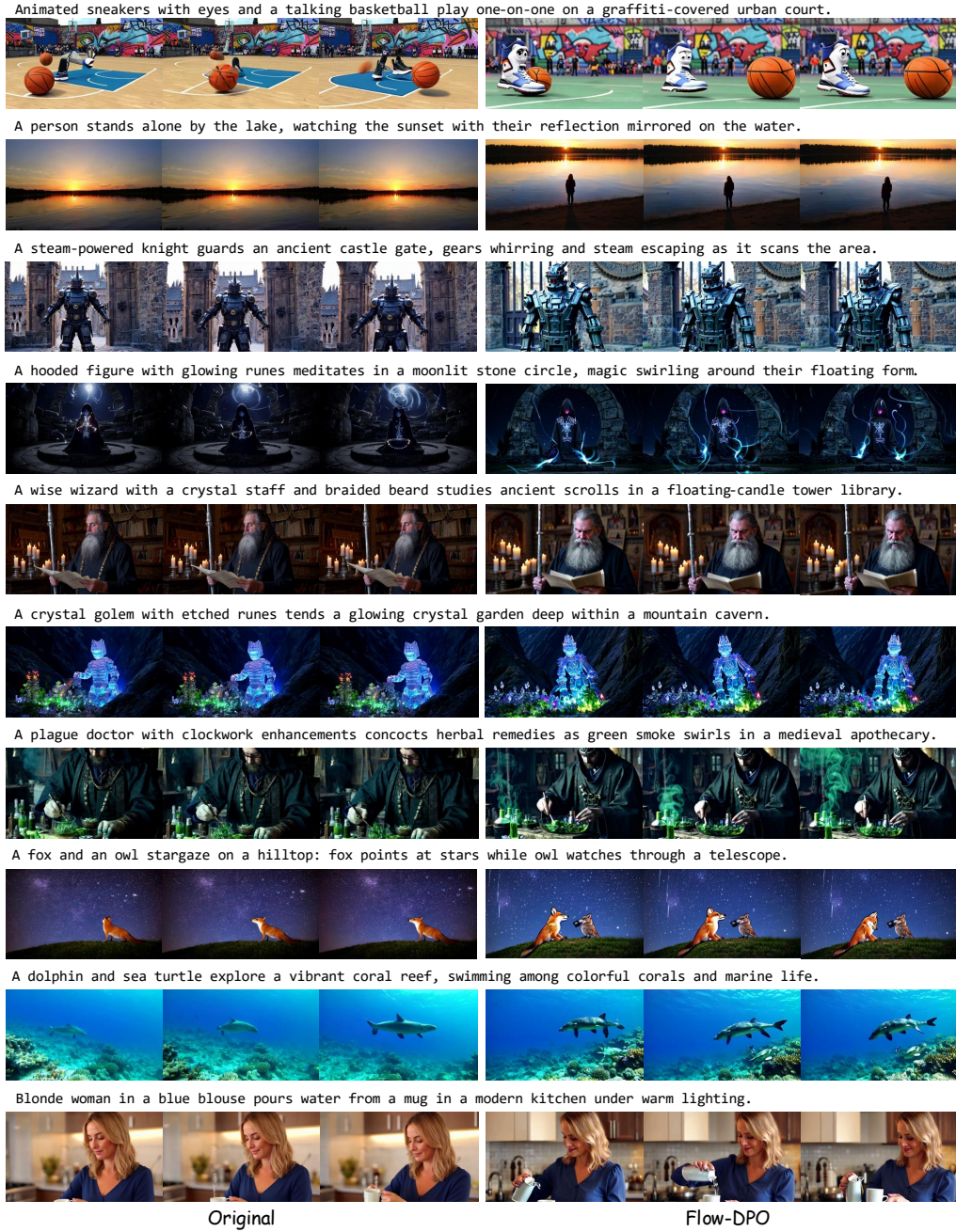


Figure 11: Additional visual comparison of videos generated by the original model and the Flow-DPO aligned model.



Figure 12: Additional visual comparison of videos generated by the original model and the Flow-DPO aligned model.

K Input Template for Reward Model

Full Input Template

[VIDEO] You are tasked with evaluating a generated video based on three distinct criteria: Visual Quality, Motion Quality, and Text Alignment. Please provide a rating from 0 to 10 for each of the three categories, with 0 being the worst and 10 being the best. Each evaluation should be independent of the others.

****Visual Quality:****

Evaluate the overall visual quality of the video, with a focus on static factors. The following sub-dimensions should be considered:

- ****Reasonableness:**** The video should not contain any significant biological or logical errors, such as abnormal body structures or nonsensical environmental setups.
- ****Clarity:**** Evaluate the sharpness and visibility of the video. The image should be clear and easy to interpret, with no blurring or indistinct areas.
- ****Detail Richness:**** Consider the level of detail in textures, materials, lighting, and other visual elements (e.g., hair, clothing, shadows).
- ****Aesthetic and Creativity:**** Assess the artistic aspects of the video, including the color scheme, composition, atmosphere, depth of field, and the overall creative appeal. The scene should convey a sense of harmony and balance.
- ****Safety:**** The video should not contain harmful or inappropriate content, such as political, violent, or adult material. If such content is present, the image quality and satisfaction score should be the lowest possible.

Please provide the ratings of Visual Quality: `<|VQ_reward|>`

END

****Motion Quality:****

Assess the dynamic aspects of the video, with a focus on dynamic factors. Consider the following sub-dimensions:

- ****Stability:**** Evaluate the continuity and stability between frames. There should be no sudden, unnatural jumps, and the video should maintain stable attributes (e.g., no fluctuating colors, textures, or missing body parts).
- ****Naturalness:**** The movement should align with physical laws and be realistic. For example, clothing should flow naturally with motion, and facial expressions should change appropriately (e.g., blinking, mouth movements).
- ****Aesthetic Quality:**** The movement should be smooth and fluid. The transitions between different motions or camera angles should be seamless, and the overall dynamic feel should be visually pleasing.
- ****Fusion:**** Ensure that elements in motion (e.g., edges of the subject, hair, clothing) blend naturally with the background, without obvious artifacts or the feeling of cut-and-paste effects.
- ****Clarity of Motion:**** The video should be clear and smooth in motion. Pay attention to any areas where the video might have blurry or unsteady sections that hinder visual continuity.
- ****Amplitude:**** If the video is largely static or has little movement, assign a low score for motion quality.

Please provide the ratings of Motion Quality: `<|MQ_reward|>`

END

****Text Alignment:****

Assess how well the video matches the textual prompt across the following sub-dimensions:

- ****Subject Relevance**** Evaluate how accurately the subject(s) in the video (e.g., person, animal, object) align with the textual description. The subject should match the description in terms of number, appearance, and behavior.
- ****Motion Relevance:**** Evaluate if the dynamic actions (e.g., gestures, posture, facial expressions like talking or blinking) align with the described prompt. The motion should match the prompt in terms of type, scale, and direction.
- ****Environment Relevance:**** Assess whether the background and scene fit the prompt. This includes checking if real-world locations or scenes are accurately represented, though some stylistic adaptation is acceptable.
- ****Style Relevance:**** If the prompt specifies a particular artistic or stylistic style, evaluate how well the video adheres to this style.
- ****Camera Movement Relevance:**** Check if the camera movements (e.g., following the subject, focus shifts) are consistent with the expected behavior from the prompt.

Textual prompt - [PROMPT]

Please provide the ratings of Text Alignment: `<|TA_reward|>`

END

L Prompt Subset of TA-Hard

A rabbit and a turtle racing on a track. The rabbit is sprinting ahead, while the turtle is steadily moving along. Spectators are cheering from the sidelines, and a finish line is visible in the distance.

A lion and a zebra playing soccer on a grassy field. The lion is dribbling the ball, while the zebra is trying to block it. The field is surrounded by trees, and other animals are watching the game.

A fox and an owl stargazing together on a hilltop. The fox is lying on its back, pointing at the stars, while the owl is perched on a nearby branch, looking through a telescope. The night sky is clear, with countless stars twinkling.

A dolphin and a sea turtle exploring a coral reef. The dolphin is swimming gracefully, while the sea turtle is gliding slowly beside it. The coral reef is vibrant with colorful corals and various marine life.

A dolphin and a whale singing together in the ocean. The dolphin is leaping out of the water, while the whale is producing deep, melodic sounds. The ocean is vast and blue, with the sun setting on the horizon.

A fox and a rabbit playing a duet on a piano in a forest clearing. The fox is playing the melody, while the rabbit is accompanying with harmony. The forest is alive with the sounds of nature, and other animals are gathered to listen.

A squirrel and a chipmunk building a treehouse in a large oak tree. The squirrel is hammering nails, while the chipmunk is holding a blueprint. The tree is tall and sturdy, with branches full of leaves.

A robot with glowing blue eyes and a human with a cybernetic arm playing basketball in a futuristic gym. The robot is dribbling the ball with precision, while the human is preparing to block the shot. The gym is equipped with advanced technology and holographic scoreboards.

A knight in shining armor and a wizard with a long, flowing beard practicing archery in a medieval courtyard. The knight is aiming at a target with a longbow, while the wizard is using magic to guide the arrows. The courtyard is surrounded by stone walls and blooming flowers.

A talking apple with eyes and a mouth, and a singing banana with legs hosting a talent show in a vibrant theater. The apple is the judge, giving feedback to contestants, while the banana is the host, entertaining the audience with jokes and songs. The theater is filled with colorful lights and excited spectators.

A pirate with a wooden leg and a mermaid with a shimmering tail playing a duet on a grand piano in an underwater cave. The pirate is playing the melody, while the mermaid is accompanying with harmony. The cave is illuminated by bioluminescent sea creatures, creating a magical atmosphere.

A superhero with a cape and a detective with a magnifying glass solving a mystery in a bustling city. The superhero is flying above the streets, scanning for clues, while the detective is examining evidence on the ground. The city is alive with activity, with skyscrapers towering overhead.

A chef with a tall hat and a robot with multiple arms cooking a gourmet meal in a state-of-the-art kitchen. The chef is chopping vegetables with precision, while the robot is simultaneously stirring, frying, and baking. The kitchen is equipped with the latest culinary technology, creating a seamless cooking experience.

A painter with a beret and a poet with a quill creating art in a sunlit studio. The painter is working on a vibrant canvas, while the poet is writing verses inspired by the artwork. The studio is filled with natural light and creative energy, with art supplies scattered around.

A spider with a square face and a green-furred puppy having a playful fight in a whimsical garden. The spider is using its web to swing around, while the puppy is playfully nipping at the spider's legs. The garden is filled with oversized flowers and colorful mushrooms.

A talking teapot with a mustache and a dancing teacup with legs performing a tea ceremony in an enchanted forest. The teapot is pouring tea, while the teacup is twirling and dancing around. The forest is magical, with glowing plants and twinkling lights.

A robot with a television screen for a head and a toaster with arms and legs having a cooking competition in a retro kitchen. The robot is displaying recipes on its screen, while the toaster is popping out perfectly toasted bread. The kitchen is styled with vintage appliances and checkered floors.

A pair of animated scissors with eyes and a mouth and a roll of tape with tiny arms and legs wrapping presents in a festive workshop. The scissors are cutting wrapping paper with precision, while the tape is sealing the packages with a smile. The workshop is decorated with holiday lights and ornaments.

A pair of animated sneakers with eyes and a mouth and a talking basketball with a face playing a game of one-on-one on an urban basketball court. The sneakers are dribbling and making quick moves, while the basketball is bouncing and trying to score. The court is surrounded by graffiti-covered walls and cheering spectators.

A paper airplane with a scarf and a paper boat with a captain's hat racing in the rain. The airplane glides through the air while the boat sails through puddles.

A basketball with a mohawk and a soccer ball with a bandana playing hopscotch in a playground. The basketball bounces high while the soccer ball rolls smoothly.

A mechanical knight with steam-powered joints standing guard at an ancient castle gate. Gears whirl softly as its head turns to scan the surroundings, while steam occasionally escapes from its armor joints.

A wandering alchemist with potion-filled vials clinking on their belt, gathering herbs in an enchanted forest where mushrooms glow and flowers whisper secrets.

A mysterious plague doctor with clockwork enhancements peeking through their dark robes, mixing herbal remedies in a medieval apothecary shop as green smoke swirls from bubbling vials.

M Broader Impacts

Our work aims to improve video generation with human feedback. While the primary goal is to improve performance, we recognize that video-generation technologies have broader societal implications. Positive applications may include enhanced creative expression and accessibility tools, thereby benefiting diverse communities. However, the same methods can be misused for creating deceptive or harmful content (e.g., deepfakes), leading to misinformation, reputational harm, and erosion of public trust.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: Yes. Our main contribution are detailed in Sec. 1.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Yes, please see Sec. 6 for limitation.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: Yes. We provide the detailed derivations and lemma proofs for Flow-DPO, Flow-RWR, and Reward Guidance in Sec 4 and Appendix B.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We detail the experimental settings in Sec 5 and evaluation settings in Appendix H. We also provide the hyperparameters in Appendix I, input template of reward model in Appendix K, pseudo-code of Flow-DPO and Flow-NRG in Appendix C. We believe our experimental results can be easily reproduced.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We include the training code for our VideoReward model in the Supplementary. The preference dataset and alignment code will be released in the future.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We detail the experimental settings in Sec 5 and evaluation settings in Appendix H. We also provide the hyperparameters in Appendix I, input template of reward model in Appendix K, pseudo-code of Flow-DPO and Flow-NRG in Appendix C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Due to the resource limitation, we do not report error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We report the GPU resources utilized during training in Appendix I.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We followed the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: See Appendix M.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [\[Yes\]](#)

Justification: We detail the human annotation process in Appendix G. We ask all annotators to manually filter out unsafe prompts during the dataset construction process.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: Yes, we credited them in appropriate ways.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: **[No]**

Justification: We include the training code for our VideoReward model in the Supplementary, along with detailed installation and usage instructions.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: **[Yes]**

Justification: We hired annotators at the standard local market rate to label the data. All annotators received training and were provided with detailed annotation guidelines (see Appendix G).

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: **[NA]**

Justification: Our study involved annotation of AI-generated videos. No personal or sensitive information was collected, and the task posed minimal risk to participants. This type of annotation work does not require IRB or equivalent approval.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.