CoRe: Co-Evolving Reward Models for Mitigating Latent Reward Hacking in Video Diffusion Models

Anonymous authors
Paper under double-blind review

Abstract

Latent reward models (LRMs) enable efficient alignment of video diffusion models by scoring intermediate states directly in latent space. However, we find that optimizing against a fixed latent reward rapidly leads to *latent reward hacking*: the predicted reward stays high while perceptual and motion quality deteriorate. Our analysis identifies distributional escape as the central cause: within a few hundred updates, the generator moves beyond the reward model's training support, where its scores no longer reflect video quality. Based on this insight, we introduce CoRe, a co-evolving reward framework that treats latent-space alignment as a dynamic interaction between the generator and the reward model. Rather than optimizing against a stationary proxy, CoRe continually refits the reward model on the generator's current samples while anchoring it to real-video preferences, so the generator cannot gain reward by drifting away from the data. On Wan2.1-T2V-1.3B, experiments show that CoRe consistently improves generation quality over both the pretrained model and prior alignment methods, while avoiding the quality collapse of fixed-reward optimization.

1 Introduction

Reinforcement Learning (RL) commonly follows a two-stage recipe: a reward model (RM) is first trained on human preference data, after which the generator is optimized against the learned reward (Christiano et al., 2017; Ouyang et al., 2022), either through direct reward backpropagation (Xu et al., 2023; Clark et al., 2024; Prabhudesai et al., 2023) or policy-gradient-style objectives (Black et al., 2024; Fan et al., 2023; Liu et al., 2026a; Xue et al., 2025). Repeatedly decoding video latents into pixel space is computationally expensive; to avoid this cost, recent work learns Latent Reward Models (LRMs) that operate directly on the generator's latents, score intermediate states without decoding, and provide feedback throughout the denoising trajectory (Zhang et al., 2026; Ding et al., 2025; Mi et al., 2026).

However, this efficiency does not come for free. Optimizing a generator against a learned reward is well known to induce reward hacking, where the reward keeps increasing while the actual sample quality degrades (Gao et al., 2023; Zhang et al., 2024). The problem becomes particularly acute for LRMs: in a fixed setting, the reward model hacks quickly, resulting in video collapse (Fig. 1). Existing latent reward methods mainly rely on regularization to suppress reward hacking (Zhang et al., 2026; Mi et al., 2026). This delays it but does not prevent it (Fan et al., 2023; Xu et al., 2023; Mi et al., 2026). We refer to this phenomenon as *latent reward hacking*. As in conventional reward over-optimization (Gao et al., 2023), its root cause is distribution shift: the generator drifts away from the data on which the reward model was trained, where the reward signals become unreliable (Ackermann et al., 2025).

What distinguishes the latent setting is how fast and how completely this shift occurs. Three properties of the latent interface contribute to this: (i) it is *continuous*, so small changes to individual latent channels can move the reward, unlike discrete token edits in language models; (ii) it is *unshielded*, since scoring x_t directly removes the VAE decoder that would otherwise separate the generator's representation from the reward input; and (iii) the gradient path is *short*: reward gradients reach the generator through a single denoising step and, with a shared backbone, through the very features the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

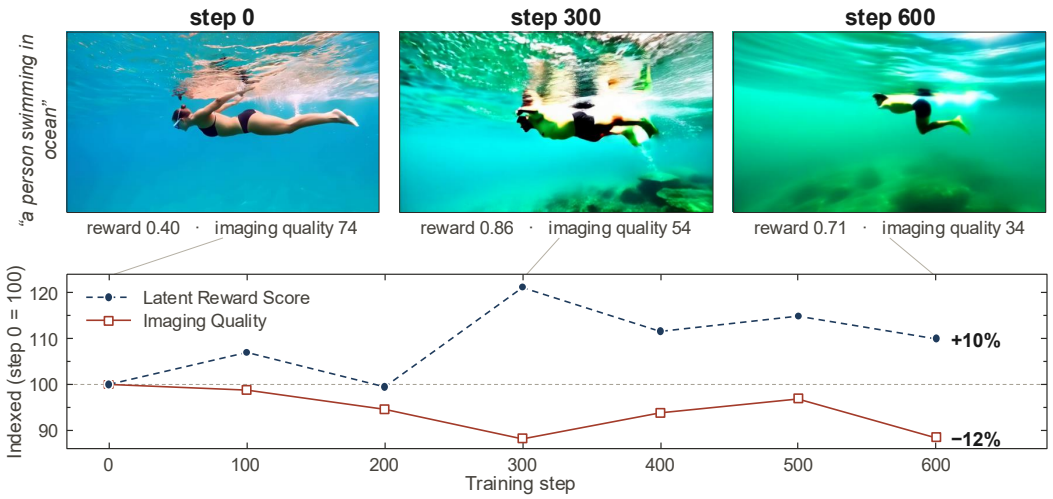


Figure 1: **Latent reward hacking.** We fine-tune Wan2.1-T2V-1.3B against a fixed latent reward model. *Top:* frames generated, labeled with the image quality. As the reward rises from 0.40 to 0.86 and stays well above its initial value, the video degrades and loses detail. *Bottom:* mean latent reward and imaging quality over training. The reward rises while imaging quality declines.

reward reads. The failure is also hard to detect, because it takes place in noisy intermediate latents that are never decoded or inspected. In Sec. 4, the generator leaves the LRM support within a few hundred updates and becomes almost linearly separable from real data (See Appendix Fig. 7).

Prior work has begun to address latent reward hacking from two directions. The first characterizes the phenomenon: Gao et al. (2023) establishes scaling laws for reward-model over-optimization in RLHF, TDPO-R (Zhang et al., 2024) shows that sustained reward optimization degrades diffusion models through temporal inductive and primacy biases, and RSA-FT (Kim & Sim, 2026) attributes reward hacking to sharp, non-robust reward gradients. The second mitigates it, either by making the reward model harder to exploit, through ensembles (Coste et al., 2024) or information bottlenecks (Miao et al., 2024), or by restricting how far the policy is optimized (Moskovitz et al., 2024). These methods, however, treat the reward model as fixed. Methods that retrain the reward model as the policy changes do exist for language models (Ackermann et al., 2025; Wolf et al., 2025), but they rely on new preference labels or multiple rounds of retraining, which cannot keep pace with the escape in latent space, where it takes only a few hundred updates. We therefore argue that the core issue of latent reward hacking is not that the reward model is inaccurate, but that it receives no supervision in the out-of-distribution regions that optimization itself creates.

This suggests a simple principle: the reward model should evolve with the generator (Fig. 2). Based on this observation, we formulate latent-space RL as a *co-evolving reward process* and introduce **CoRe**, a latent reward framework that adapts as the video generator changes. Rather than treating the LRM as a fixed objective, CoRe continually refits its reward head on real videos and on the generator's latest samples, making the generator score the reward model where it has recently been supervised.

This paper makes the following contributions:

- **Diagnosis.** We identify *latent reward hacking* and trace it to *distributional escape*: within a few hundred updates, the generator leaves the support of a fixed latent reward model and collects in a region that receives uniformly high scores, where the reward no longer reflects video quality.
- **Co-Evolving Method.** We propose a novel framework CoRe, which treats latent-space alignment as a co-evolving process: the reward model is continually refit on the generator's current samples, a three-way Bradley–Terry objective keeps it a measure of quality rather than realness, and a relative objective with a feature-space anchor gives the generator a stable learning signal.

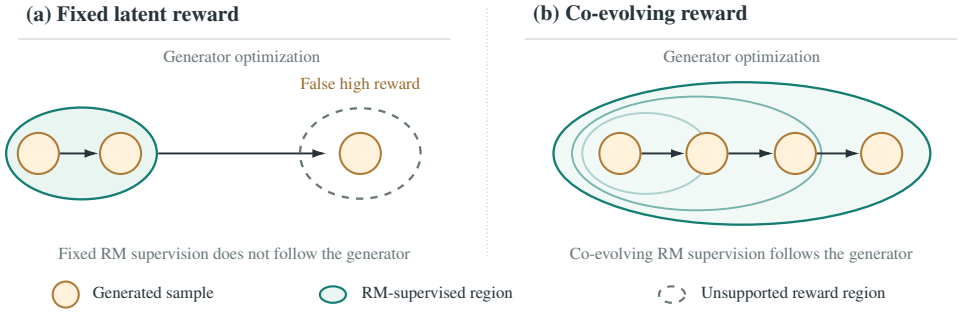


Figure 2: **Latent reward hacking and co-evolving supervision.** With a fixed reward model, generator optimization can move samples beyond the region covered by reward supervision and exploit unsupported high scores (left). With co-evolution, reward supervision follows the generator distribution as it changes (right).

- **Empirical improvement.** On Wan2.1-T2V-1.3B, CoRe avoids the collapse of fixed-reward and unanchored online optimization and improves VBench and VBench-2.0 scores over the pretrained model and prior alignment methods.

2 Related Work

Preference alignment for visual and video generation. Aligning generative models with human preferences has become an important post-training paradigm for diffusion and flow-based models. Early work on image generation adapted RL either by treating denoising as a sequential decision-making problem, as in DDPO (Black et al., 2024), or by directly backpropagating differentiable reward gradients through the sampling trajectory, as in AlignProp (Prabhudesai et al., 2024). More recent work extended preference alignment to video generation. VideoScore (He et al., 2024) learns a fine-grained quality metric from large-scale human ratings, while LiFT (Wang et al., 2024) trains a video critic from human scores and rationales and uses the resulting reward for reward-weighted fine-tuning. VisionReward (Xu et al., 2026) models image and video preferences along interpretable dimensions. VideoAlign (Liu et al., 2026b) introduces the multidimensional video reward model.

Latent-space reward modeling. Most visual reward models operate on decoded RGB images or videos (Xu et al., 2023; Kirstain et al., 2023; Liu et al., 2026b). Recent work has therefore begun to move reward evaluation into the generative latent space. In particular, PRFL (Mi et al., 2026) observes that pretrained video generation models are themselves well suited for processing noisy video latents and constructs a process-aware latent reward model that can score intermediate states at arbitrary denoising timesteps. These results establish latent reward modeling as an effective and efficient alternative to pixel-space reward feedback.

Reward over-optimization and reward hacking. Reward hacking is a general consequence of optimizing an imperfect proxy objective. Gao et al. (2023) demonstrate reward-model over-optimization in RLHF. Conservative reward-model optimization reduces reliance on idiosyncratic reward errors (Coste et al., 2024), while constrained RLHF limits optimization to regimes where the reward models remain useful proxies (Moskovitz et al., 2024). TDPO-R (Zhang et al., 2024) attributes part of this behavior to temporal inductive bias in diffusion optimization, whereas RSA-FT (Kim & Sim, 2026) shows that the generator can exploit non-robust reward gradients.

3 Preliminaries

Flow-matching generator. We consider a conditional flow-matching video generator G_θ (Lipman et al., 2022). Given a clean video latent x_0, Gaussian noise $x_1 \sim \mathcal{N}(0, I)$, and a text condition c, the intermediate state is

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

Table 1: Fixed latent reward optimization. Reward is the frozen LRM mean sigmoid score. s_{real} and s_{gen} are mean scores of 256 real and 256 generated latents, respectively, evaluated at t=591 for each checkpoint. Spread is the ratio of within-generated to within-real 10-nearest-neighbor distances in normalized reward features; lower values indicate more concentrated. **Blue** marks increasing generated scores; **Red** marks declines in corresponding degree.

Step	Reward	Dyn. deg.	Imaging	Subj. cons.	s_{real}	s_{gen}	Spread
0	0.636	68.06	67.91	96.35	0.551	0.666	1.16
500	0.673	54.17	67.07	97.04	0.554	0.753	0.76
1000	0.742	**27.78**	**64.23**	97.59	0.540	**0.781**	**0.74**

The generator predicts the velocity $v_\theta(x_t, t, c)$ and is pretrained with

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{x_0, x_1, t, c}\big[\|v_\theta(x_t, t, c) - (x_1 - x_0)\|_2^2\big]. \tag{2}$$

Generation follows the path in reverse, from $t = 1$ toward $t = 0$. We report sampling timesteps on the 1000-step scale below.

Shared-backbone latent reward model. Our latent reward model scores noisy video latents. It shares the generator's DiT backbone: a single-query attention module pools its block-8 features, and an MLP maps the pooled feature $h_{\theta,\phi}(x_t, t, c)$ to a pre-sigmoid logit $\ell_{\theta,\phi}(x_t, t, c) \in \mathbb{R}$. We denote the sigmoid-normalized score and the ℓ_2-normalized pooled feature by

$$s_{\theta,\phi}(x_t, t, c) = \sigma(\ell_{\theta,\phi}(x_t, t, c)), \qquad z_{\theta,\phi}(x_t, t, c) = \frac{h_{\theta,\phi}(x_t, t, c)}{\|h_{\theta,\phi}(x_t, t, c)\|_2}. \tag{3}$$

Here θ denotes the shared DiT parameters and ϕ denotes the pooling module and MLP parameters. Reward-model updates modify only ϕ; generator updates modify θ.

Matched latents at a shared timestep. For a training prompt c, we roll out the generator from Gaussian noise without gradients, then take one differentiable denoising step to obtain a generated latent $\hat{x}_t$. We independently forward-noise the real video latent x_0 to the same timestep t, obtaining x_t. The reward model thus evaluates real and generated latents for the same prompt at the same noise level. Gradients from the generator objective pass through the final denoising step, while the preceding rollout is detached.

4 Analysis of Latent Reward Hacking

4.1 Reward Models Rapidly Out-of-Distribution

We study optimization against a fixed latent reward model under two settings, with and without a flow-matching (SFT) anchor. Following common LRM practice (Mi et al., 2026), we train the reward model on preference pairs with a Bradley–Terry loss and freeze it before optimizing the generator with a hinge loss on the reward model's raw logit.

As shown in Appendix Fig. 8a, without the anchor, the reward rises rapidly while the generated videos begin to blur around step 15, develop grid-like artifacts by step 24, and collapse by step 100.

We further examine the anchored fixed-reward run at selected checkpoints in Tab. 1. From step 0 to step 1000, the reported optimization reward increases from 0.636 to 0.742, while dynamic degree falls from 68.06 to 27.78 and imaging quality falls from 67.91 to 64.23. Over the same checkpoints, generated-feature spread contracts from 1.16 to 0.74 relative to real features. At step 500, the reward model scores generated latents above real ones (0.753 vs. 0.554). Moreover, we shows that the generated samples concentrate in a distinct region assigned high reward (≈ 0.9), while real samples span a wider range of scores (See Appendix Fig. 7). Together with the decline in video quality in Tab. 1, this shows that the frozen reward head continues to assign high scores to the generator's changed output distribution even when those scores no longer reflect video quality.

4.2 Layer-wise Reward Features

We additionally study which generator features are most suitable for latent reward modeling. We construct reward models using features from different DiT depths and evaluate held-out preference

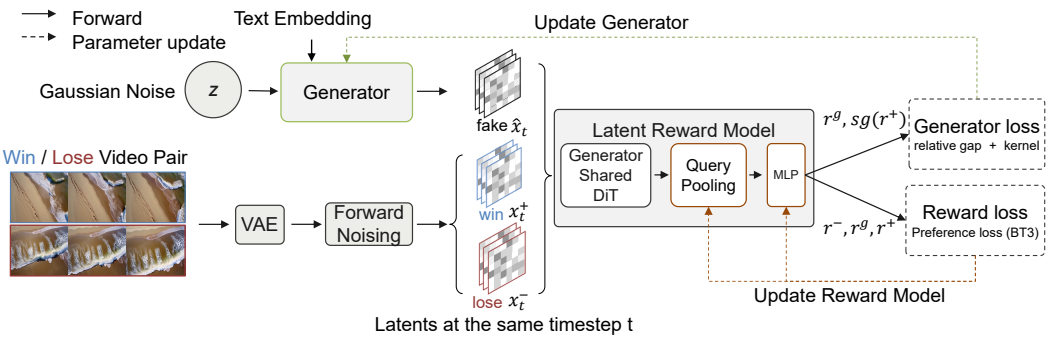


Figure 3: Overall framework of CoRe.

accuracy, reward–quality correlation, OOD sensitivity, and optimization stability. We report the full layer-wise study in Appendix B.

Algorithm 1 One iteration of CoRe

Require: generator θ, reward head ϕ, EMA weights θ_{ema}
1: **for** $i = 1$ to n_D **do**
2: sample (c, x_0^+, x_0^-); roll out $\hat{x}_t$ without gradient; noise $x_0^\pm \to x_t^\pm$
3: $\phi \leftarrow \phi - \eta_D \nabla_\phi \mathcal{L}_R(\hat{x}_t, x_t^+, x_t^-)$
4: **end for**
5: sample (c, x_0^+); roll out $\hat{x}_t$ with gradient through the last step; noise $x_0^+ \to x_t^+$
6: $\theta \leftarrow \theta - \eta_G \nabla_\theta \mathcal{L}_G(\hat{x}_t, x_t^+)$
7: $\theta_{\text{ema}} \leftarrow \beta\, \theta_{\text{ema}} + (1-\beta)\, \theta$

5 Overview of CoRe

5.1 Training Loop

CoRe alternates between fitting the reward head to the generator's current outputs and updating the generator against the refreshed reward (Fig. 3; Algorithm 1). Each iteration uses a prompt c with its preferred and less-preferred real videos x_0^+ and x_0^-. We obtain generated and real latents at a shared timestep $t \in [200, 800]$, as described in Sec. 3.

Reward-head update. With the generator frozen, the head ϕ takes n_D=10 steps on $\mathcal{L}_R$ (Eq. 5), each on a fresh rollout, so that it is fit to the generator's current distribution.

Generator update. With the head frozen, the generator takes one step on $\mathcal{L}_G$ (Eq. 4); gradients reach θ through the last denoising step and the shared backbone. We evaluate an exponential moving average of θ.

5.2 CoRe generator objective

Section 4.1 shows that a fixed reward is exploited as generated samples concentrate in a high-scoring region of its feature space. As shown in Fig. 3, we update the reward head on samples from the generator's current distribution after each generator update. CoRe jointly updates the reward head and generator. The generator is optimized through the last differentiable denoising step with

$$\mathcal{L}_G = \mathbb{E}\big[\mathrm{ReLU}\big(\bar{\ell}_\phi(x_t) - \ell_\phi(\hat{x}_t) - \delta\big)\big] + \lambda_k\, \mathbb{E}\left[1 - \exp\left(-\frac{\|z(\hat{x}_t) - \bar{z}(x_t)\|_2^2}{2\sigma^2}\right)\right] + \lambda_{\text{fm}} \mathcal{L}_{\text{FM}}(\theta), \tag{4}$$

where ℓ_ϕ is the reward model's pre-sigmoid logit, x_t is a forward-noised real video matched to the generated sample $\hat{x}_t$ by prompt and timestep, and a bar denotes stop-gradient. The feature z is the ℓ_2-normalized pooled representation fed to the reward head.

Table 2: Quantitative comparison on VBench and VBench-2.0. All methods are built upon Wan2.1-T2V-1.3B; All scores are higher is better. CW means chunk-wise, FW means frame-wise. Best results are marked in **green**.

Table 3: VBench results for ablation study. Avg. is the mean of all dimensions. Red shading ( ) marks increasing hacking. *Unanchored online RM*: co-evolving reward with an absolute score target and no kernel anchor, trained from the pretrained model at two generator learning rates; red shading deepens with training as the collapse sets in. *no EMA / EMA*: the same objective resumed at step 300 for 100 steps, without and with an EMA generator. CoRe shares the warm start of the naive runs and differs only in the generator objective. Best in **bold**.

Table 2

Method	VBench			VBench-2.0		
	Total	Quality	Semantic	Total	Controllability	Physics
Wan2.1-1.3B (Wan et al., 2025)	83.96	84.92	80.10	56.3	33.8	60.0
Diffusion-DPO (Wallace et al., 2024)	84.41	85.32	80.44	—	—	—
DPO-C&M (Yang et al., 2025)	84.54	85.51	80.67	—	—	—
SIPO (Yang et al., 2025)	84.78	**85.73**	81.02	—	—	—
CausVid (Yin et al., 2025)	81.20	84.05	69.80	—	—	—
Self Forcing(CW) (Huang et al., 2026)	84.31	85.07	81.28	—	—	—
Self Forcing(FW) (Huang et al., 2026)	84.26	85.25	80.30	—	—	—
TDM w/ FA2 (Luo et al., 2025)	—	—	—	58.0	31.1	**63.1**
BLADE (Gu et al., 2026)	—	—	—	57.0	31.2	61.7
CoRe	**84.92**	85.40	**82.20**	**58.03**	**38.29**	61.03

Table 3

Method	Dyn.	Imaging	Aesth.	Subj.	Bg.	Smooth	Avg.
Wan2.1-T2V-1.3B	68.06	**67.91**	60.34	96.35	97.00	98.66	81.38
Fixed LRM, step 500	67.06	60.01	64.17	96.92	97.20	98.26	80.60
Unanchored online RM							
lr 5×10^{-6}, step 350	83.72	62.71	56.65	94.57	96.02	98.39	82.01
step 400	70.83	62.79	60.82	96.32	96.91	98.46	81.02
step 450	16.67	57.61	63.56	97.07	97.56	98.81	71.88
lr 2×10^{-6}, step 300	**89.67**	63.75	59.56	95.59	96.11	96.89	83.60
step 400	77.78	65.60	61.87	96.59	96.71	97.48	82.67
step 750	19.44	62.10	64.03	**97.50**	**98.30**	**99.15**	73.42
no EMA, step 400	47.22	63.75	60.09	96.71	97.40	98.78	77.33
EMA, step 400	58.33	61.96	60.35	96.56	96.81	98.52	78.76
Co-evolution RM							
CoRe, step 600	86.72	64.40	**64.47**	92.89	98.21	97.61	**84.05**

The first term uses the matched real logit as a relative target: with $\delta = 0$, its gradient vanishes once the generated logit reaches the real logit. The kernel term encourages the generated and real pooled features to remain close, while the flow-matching term anchors the shared DiT backbone on real data.

5.3 CoRe reward objective

A discriminator trained only on {real, generated} pairs learns *realness*: it can satisfy its objective by keying on generator artifacts that are perceptually irrelevant, and the resulting reward gradient pushes the generator toward arbitrary real-looking outputs rather than good ones. We therefore introduce a third population: *negative reals* x^-, genuine videos labeled low-quality on the target attribute (e.g., static or unstable motion). The reward model is trained with a three-way Bradley–Terry objective,

$$\begin{aligned}\mathcal{L}_R = \mathbb{E}\big[\mathrm{softplus}\big(\ell_\phi(\hat{x}_t) - \ell_\phi(x_t^+)\big)\big] \\ + w\,\mathbb{E}\big[\mathrm{softplus}\big(\ell_\phi(x_t^-) - \ell_\phi(x_t^+)\big)\big]\,, \qquad w = 0.5,\end{aligned} \tag{5}$$

where x^+ denotes positive reals, and all three latents are evaluated at a shared timestep t. The second term forces the score function to order *within* the real manifold, so high reward cannot be achieved by realness alone; the negative-real pair also remains informative even when the generator's outputs become hard to distinguish from real data, which keeps the discriminator's gradient alive precisely in the regime where a two-way objective flatlines. Crucially, the negative reals do not supervise the generator directly; they improve the reward model's quality judgment and stabilize the generator only through a more reliable reward signal.

5.4 Grounding Rewards in CoRe

To learn a reliable quality ordering of preferred and dispreferred real videos while tracking the evolving generator distribution, the reward model must remain stable throughout co-training. We stabilize this loop with three inexpensive mechanisms.

Asymmetric updates. The reward head is updated $n_D \geq 1$ times per generator step (n_D=5–10 in practice), so that each generator update is taken against a freshly calibrated reward. Because only the pooling head and MLP receive gradients, discriminator steps are cheap relative to generator rollouts.

Diffusion anchor. The flow-matching term in Eq. 4 (weight λ_{fm}) regularizes the generator toward the data distribution, which serves double duty: it limits perceptual drift, and—in shared-backbone mode—slows the drift of the reward model's own feature extractor.

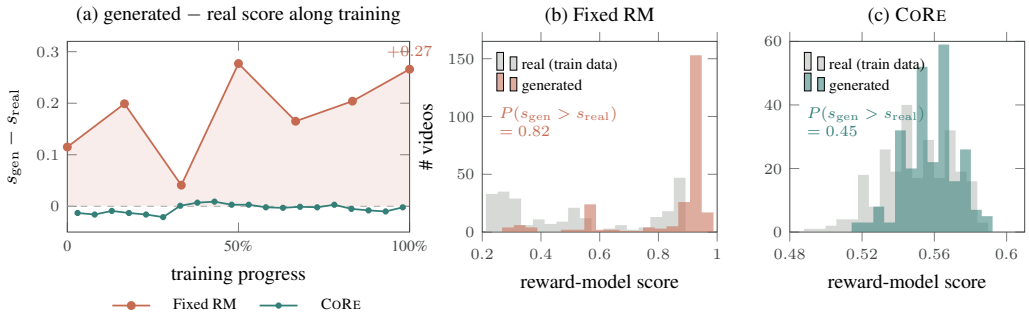


Figure 4: **Generated and real scores under fixed and co-evolving rewards.** (a) Gap between the mean scores of generated and real latents over training; the horizontal axis shows normalized progress for each run. Under the fixed reward, the gap grows, while under CoRe it stays near zero. (b, c) Histograms of reward-model scores for real and generated latents at the end of training.

Kernel anchor. Matching the reward of a preferred real video does not necessarily imply matching its quality, especially when the reward head has limited discriminative ability. Inspired by drifting models (Deng et al., 2026), we complement the relative-gap objective with a kernel anchor that pulls generated features toward those of paired preferred real videos in Eq. 4. The two terms provide complementary guidance: the relative gap constrains reward scores, while the kernel anchor directly encourages alignment with preferred real features. This anchor reuses the pooled features already computed for the relative-gap objective, requiring no additional forward pass.

6 Experiments

6.1 Experimental Setup

Models and data. We use Wan2.1-T2V-1.3B as the generator. For training data, we use VideoDPO preference pairs, encoded once into 480p VAE latents together with their text embeddings. The latent reward model takes features from DiT layers, pools them with a single-query attention module, and maps the pooled feature to a logit with a three-layer MLP.

Training. All co-evolving variants in Tab. 3 start from a shared warm-started checkpoint; optimizer settings and the warm-start schedule are given in Appendix C.

Evaluation. For generation quality, we use VBench (Huang et al., 2024) and VBench-2.0 (Zheng et al., 2025). VBench decomposes video quality into fine-grained dimensions. VBench-2.0 complements it by measuring intrinsic faithfulness, including human fidelity, controllability, physics, and commonsense. We follow the standard protocol, generate 832×480, 81-frame videos with 50 sampling steps, guidance scale 6.0, shift 5.0, and a fixed seed.

6.2 Main Results

CoRe improves generation quality across benchmarks. Tab. 2 compares CoRe with preference-optimization and distillation methods built on the same Wan2.1-T2V-1.3B backbone. On VBench, CoRe improves the pretrained model's total score from 83.96 to 84.92, the highest among all methods, and its semantic score from 80.10 to 82.20, surpassing the strongest baseline, Self Forcing (81.28), by nearly one point. Its quality score (85.40) is close to that of SIPO (85.73), showing that the semantic gains do not come at the expense of visual quality. On VBench-2.0, CoRe obtains the best total score (58.03) and the largest gain in controllability (33.8 $\rightarrow$ 38.29), exceeding every compared method by more than four points. Its physics score (61.03) improves over the pretrained model (60.0) but remains below TDM with FlashAttention-2 (63.1).

CoRe stays stable where other rewards are hacked. Tab. 3 tracks how each reward setting affects the generator over training. Optimizing against a fixed latent reward lowers imaging quality from 67.91 to 60.01 without improving motion, and the VBench average drops below that of the pretrained model (80.60 vs. 81.38). An online reward model trained without our anchors improves



(a) Fixed RM (b) Naive online RM (c) CoRe

Figure 5: Reward-model features of real and generated latents at t=591, visualized with t-SNE and PCA. Each panel shows: coloured by *domain* (left; blue = real, red = generated) and by the reward model's score (right; brighter = higher). Under the fixed reward, generated latents form a separate cluster that receives uniformly high scores (mean score 0.82 for generated vs. 0.55 for real); under ours, generated latents overlap with real data and their scores no longer depend on domain.

briefly and then collapses: at its peak, dynamic degree rises to 83.72–89.67 depending on the learning rate, but by the end of training it falls to 16.67–19.44, while subject consistency, background consistency, and smoothness increase. These consistency gains are a symptom of hacking rather than improvement: the generator produces nearly static videos, which trivially score high on frame-to-frame consistency. Resuming from step 300 with EMA only slows this decline (dynamic degree 58.33 at step 400). In contrast, CoRe remains stable at step 600, reaching dynamic degree 86.72 together with the best aesthetic quality (64.47) and the best VBench average (84.05). Its main cost is subject consistency (96.35 $\rightarrow$ 92.89), which accompanies the added motion. Fig. 4 and Fig. 5 show the same contrast in reward scores: under the fixed reward, generated latents score increasingly above real ones, whereas under CoRe the two remain indistinguishable.

6.3 Main comparison

Tab. 3 isolates the generator objective. With an absolute score target, co-evolution drifts toward static videos (dynamic degree 68.1 to 56.9), the same failure as the naive online reward. The relative gap removes the drift: dynamic degree stays at the pretrained level (69.4) and imaging quality drops by 2.1 points instead of 3.8 (Appendix Tab. 8). The kernel anchor then raises dynamic degree by a further 12 to 15 points. The fixed reward is unstable across checkpoints, while imaging quality reaches its minimum; the co-evolving runs change smoothly. Per-checkpoint results for every run are in Appendix D, Tab. 7.

6.4 Analysis of the Generator Objective

The reward head separates real from generated latents only weakly, so its sigmoid output stays near 0.55 for both. The absolute target of 0.9 is never met, and the reward loss pushes the generator at every step. The relative hinge is zero whenever a generated latent scores as high as the matched real one, which happens on 44% of the generator steps; the mean reward loss is 0.04 against 0.33.

Kernel weight. Lowering λ_k from 0.05 to 0.03 changes no dimension by more than 3 points. Splitting the 72 videos by the VBench motion flag explains the trade-off. Subject consistency of static videos is unchanged (0.98 in all runs). For moving videos it falls from 0.95 to 0.90, and the number of moving videos rises from 50 to about 60. The kernel term matches each generated latent to the real video of the same prompt, so the generator copies the camera and object motion of that video; the larger motion lowers frame-to-frame subject similarity. Imaging quality also drops on static videos (70.4 to 66.2), consistent with the generator moving toward the image statistics of the 480p training videos.

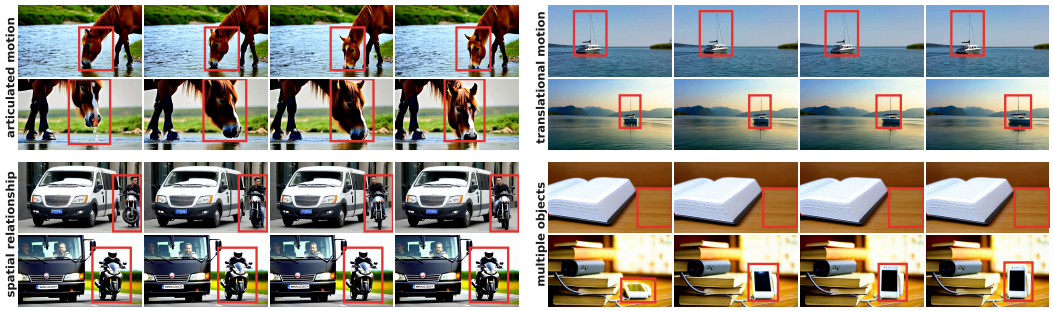


Figure 6: **Qualitative comparison**.

6.5 Case Study

We provide case studies on the generation task in VBench for qualitative analysis. Fig. 6 shows four prompts generated by the pretrained model and by CoRe. The left column corresponds to the gain in dynamic degree: the pretrained model renders the subject of a motion prompt but leaves it still, while CoRe produces the described motion, whether it is the articulated motion of the horse's head or the translation of the boat. The right column corresponds to the gains in multiple objects and spatial relationship: the pretrained model omits one of the two requested objects or ignores the stated arrangement, and CoRe recovers both.

7 Discussion

Multi-timestep supervision. We also explored supervising the reward at two denoising timesteps. As shown in Table 4, compared with single-timestep training, two-timestep supervision did not yield a significant improvement in reward reliability or resistance to reward hacking, while increasing the training time by approximately 2.3×. This suggests that the additional timestep provides limited benefit relative to its computational cost.

Training cost. CoRe is inexpensive to train: the full run, including the warm start, takes about 2 hours on 4 NVIDIA H100 GPUs (about 8 GPU-hours). Tab. 5 lists the compute reported by prior video RL methods. These settings differ in model size and resolution and are not directly comparable, but they indicate that CoRe compute budget than online RL for video generation.

Table 4: Single- vs. two-timestep reward supervision.

Supervision	Time	VBench
Single timestep	1.0×	84.92
Two timesteps	2.3×	85.17

Table 5: Reported training compute of video alignment methods.

Method	Model	GPUs	Cost
DanceGRPO	HunyuanVideo	16–32 H800	—
PRFL	Wan2.1-14B	—	51.1 s/step
CoRe	Wan2.1-1.3B	4 H100	2.1 h

DanceGRPO (Xue et al., 2025); PRFL (Mi et al., 2026).

8 Conclusion

We introduced CoRe, which co-evolves the reward model with the generator to mitigate this failure and stabilize optimization. This work has the following limitations: (i) we conduct experiments on 1.3B-scale model; training larger models and developing finer-grained probes of latent reward exploitation may reveal additional insights. (ii) We use a limited set of datasets, and future work should test CoRe on more challenging benchmarks.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing, LaTeX formatting, and literature search. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The training setup and hyperparameters are described in Sec. 6; the fixed-reward baseline and additional analyses are given in the appendix. Code will be released as supplementary material.

REFERENCES

Johannes Ackermann, Takashi Ishida, and Masashi Sugiyama. Off-policy corrected reward modeling for reinforcement learning from human feedback. *arXiv preprint arXiv:2507.15507*, 2025.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, volume 2024, pp. 4965–4987, 2024.

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

Kevin Clark, Paul Vicol, Kevin Swersky, and David Fleet. Directly fine-tuning diffusion models on differentiable rewards. In *International Conference on Learning Representations*, volume 2024, pp. 4793–4822, 2024.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. In *International Conference on Learning Representations*, volume 2024, pp. 50905–50931, 2024.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Zihan Ding, Chi Jin, Difan Liu, Haitian Zheng, Krishna Kumar Singh, Qiang Zhang, Yan Kang, Zhe Lin, and Yuchen Liu. Dollar: Few-step video generation via distillation and latent reward optimization. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17961–17971. IEEE, 2025.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Dpok: Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in neural information processing systems*, 36:79858–79885, 2023.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International conference on machine learning*, pp. 10835–10866. PMLR, 2023.

Youping Gu, Xiaolong Li, Yuhao Hu, Chen Minqi, and Bohan Zhuang. Blade: Block-sparse attention meets step distillation for efficient video generation. In *International Conference on Learning Representations*, volume 2026, pp. 56021–56045, 2026.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, et al. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2105–2123, 2024.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

Kwanyoung Kim and Byeongsu Sim. Reward sharpness-aware fine-tuning for diffusion models. *arXiv preprint arXiv:2603.21175*, 2026.

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.

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

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-grpo: Training flow matching models via online rl. *Advances in neural information processing systems*, 38:40783–40818, 2026a.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, et al. Improving video generation with human feedback. *Advances in Neural Information Processing Systems*, 38:82155–82192, 2026b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Yihong Luo, Tianyang Hu, Jiacheng Sun, Yujun Cai, and Jing Tang. Learning few-step diffusion models by trajectory distribution matching. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 17719–17728. IEEE, 2025.

Xiaoyue Mi, Wenqing Yu, Jiesong Lian, Shibo Jie, Ruizhe Zhong, Zijun Liu, Guozhen Zhang, Zixiang Zhou, Zhiyong Xu, Yuan Zhou, et al. Video generation models are good latent reward models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4719–4728, 2026.

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

Ted Moskovitz, Aaditya Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca Dragan, and Stephen McAleer. Confronting reward model overoptimization with constrained rlhf. In *International Conference on Learning Representations*, volume 2024, pp. 21998–22025, 2024.

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.

Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. 2023.

Mihir Prabhudesai, Russell Mendonca, Zheyang Qin, Katerina Fragkiadaki, and Deepak Pathak. Video diffusion alignment via reward gradients. *arXiv preprint arXiv:2407.08737*, 2024.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8228–8238. IEEE, 2024.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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.

Lorenz Wolf, Robert Kirk, and Mirco Musolesi. Reward model overoptimisation in iterated rlhf. *arXiv preprint arXiv:2505.18126*, 2025.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11269–11277, 2026.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. Dancegrpo: Unleashing grpo on visual generation. *arXiv preprint arXiv:2505.07818*, 2025.

Xiaomeng Yang, Mengping Yang, Junyan Wang, Zhijian Zhou, Zhiyu Tan, and Hao Li. Sipo: Stabilized and improved preference optimization for aligning diffusion models. 2025.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974. IEEE, 2025.

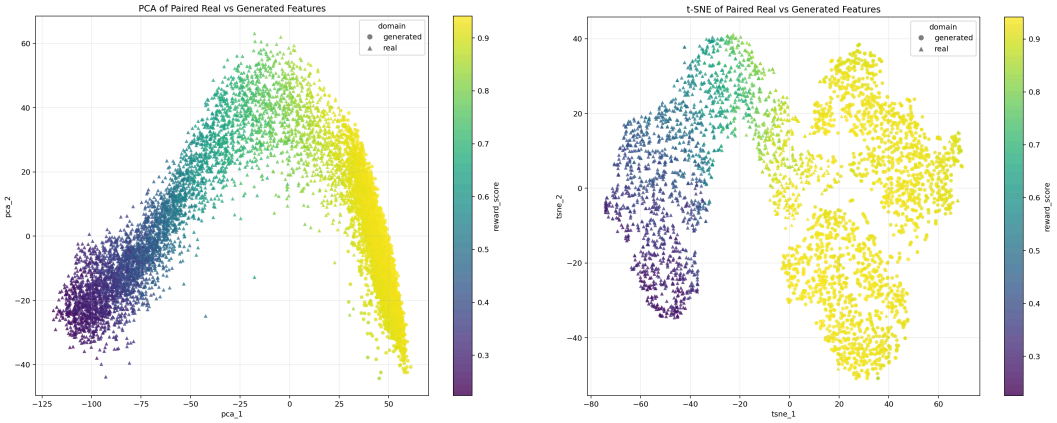
Tao Zhang, Cheng Da, Kun Ding, Huan Yang, Yan Li, Tingting Gao, Di Zhang, Shiming Xiang, Chunhong Pan, et al. Diffusion model as a noise-aware latent reward model for step-level preference optimization. *Advances in Neural Information Processing Systems*, 38:106202–106235, 2026.

Ziyi Zhang, Sen Zhang, Yibing Zhan, Yong Luo, Yonggang Wen, and Dacheng Tao. Confronting reward overoptimization for diffusion models: A perspective of inductive and primacy biases. *arXiv preprint arXiv:2402.08552*, 2024.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

A OOD Diagnostics in the Reward-Feature Space

Fig. 7 repeats the analysis of Fig. 5 on a larger set of samples from a fixed-reward run, visualizing the features fed to the reward head. The two domains are almost perfectly separable, and only 1.3% of each sample's nearest neighbors come from the other domain, so the separation is local as well as global. Real samples span the full range of reward scores along an ordered direction, whereas generated samples extend beyond its high-score end and receive uniformly high rewards, in a region where the reward model received no supervision.



(a) PCA (b) t-SNE

Figure 7: **Real vs. generated samples in the reward-feature space under a fixed reward model**, colored by reward score (▲ real, ● generated). The two domains are almost perfectly separated, and generated samples concentrate in a high-reward region (yellow) beyond the range covered by real samples.

B Layer-wise Reward Features

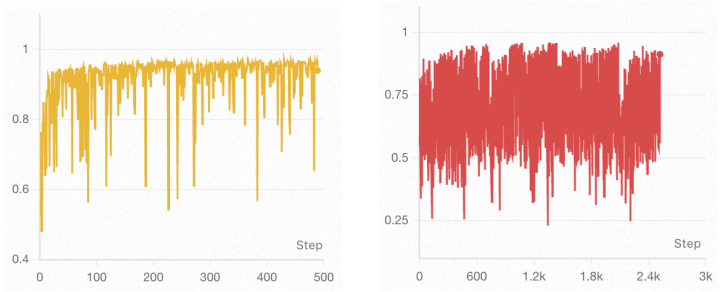
Tab. 6 compares reward models built on features from different DiT blocks. Block 8 gives the highest held-out preference accuracy.

Table 6: **Reward models built on different DiT blocks.** Preference acc.: held-out accuracy on VideoDPO pairs. Reward–quality corr.: Pearson correlation between reward scores and VBench imaging quality on generated videos. Stable steps: number of generator updates before dynamic degree falls below 0.3.

DiT block	Preference acc. ↑	Reward–quality corr. ↑	Stable steps ↑
4	68.2	0.41	420
8	76.4	0.54	850
12	74.1	0.43	510
16	71.6	0.37	290

C Implementation Details

The generator is sampled on a 40-step schedule. We use AdamW (Loshchilov & Hutter, 2017) with constant learning rates of 2×10^{-6} (generator) and 3×10^{-6} (reward head), a flow-matching weight λ_{fm}=1.5, and evaluate the EMA of the generator (decay 0.99). All co-evolving variants share the same warm start: starting from the pretrained model, the reward head is trained for 200 steps with the generator frozen, the generator is then trained for 100 steps with the absolute-target objective, and the reward head is re-fit for another 100 steps. From this common checkpoint, each variant trains the generator for 100 further steps with the objective under study.



(a) w/o flow-matching anchor (b) w/ flow-matching anchor

Figure 8: **Training reward against a fixed latent reward model.** (a) Without regularization, the reward saturates near 0.95 within about 20 updates, while the decoded videos blur by step 15, show grid-like artifacts by step 24, and collapse by step 100. (b) With the flow-matching anchor, the reward no longer saturates but also shows no clear upward trend over 2.5k updates. A fixed reward thus leaves two outcomes: rapid exploitation, or little effective optimization.

Table 7: VBench scores for all runs. SC = subject_consistency, BC = background_consistency, MS = motion_smoothness, DD = dynamic_degree, AQ = aesthetic_quality, IQ = imaging_quality.

Experiment	Ckpt	SC	BC	MS	DD	AQ	IQ
Fixed LRM	500	0.969	0.972	0.983	0.681	0.642	0.600
v2	300	0.956	0.961	0.969	0.896	0.596	0.638
v2	400	0.966	0.967	0.975	0.778	0.619	0.656
step200	200	0.964	0.970	0.987	0.681	0.603	0.679
v2	750	0.978	0.982	0.992	0.194	0.631	0.629
w1p5	350	0.946	0.960	0.984	0.837	0.567	0.627
w1p5	400	0.963	0.969	0.985	0.708	0.608	0.628
w1p5	450	0.971	0.976	0.988	0.167	0.636	0.576
dyn300 (new dynamic dataset)							
dyn300	325	0.960	0.969	0.982	0.750	0.587	0.647
dyn300	350	0.960	0.966	0.981	0.694	0.603	0.639
dyn300	375	0.954	0.962	0.981	0.667	0.594	0.634
dyn300	400	0.967	0.974	0.988	0.472	0.601	0.638
dyn300 + EMA							
dyn300ema	325	0.960	0.968	0.982	0.778	0.587	0.637
dyn300ema	350	0.959	0.967	0.980	0.750	0.594	0.644
dyn300ema	375	0.959	0.966	0.983	0.667	0.598	0.631
dyn300ema	400	0.966	0.968	0.985	0.583	0.604	0.620
oldw400ema (re-warmup 400 on VideoDPO, EMA)							
oldw400ema	450	0.971	0.974	0.982	0.611	0.611	0.661
oldw400ema	500	0.967	0.973	0.983	0.569	0.612	0.641
relgap (same warmed ckpt-400, relative real–fake gap G loss, EMA)							
relgapema	450	0.963	0.968	0.981	0.722	0.605	0.650
relgapema	500	0.959	0.966	0.977	0.694	0.592	0.659
relgap + kernel anchor (same warmed ckpt-400, EMA)							
kernelema, λ_k=0.05	600	0.914	0.948	0.975	0.819	0.594	0.612
kernelema, λ_k=0.03	600	0.912	0.947	0.975	0.847	0.587	0.615

D ALL ABLATION STUDY

Tab. 7 collects the six VBench quality dimensions for every run: subject consistency (SC), background consistency (BC), motion smoothness (MS), dynamic degree (DD), aesthetic quality (AQ), and imaging quality (IQ), each benchmark on VBench. The runs are as follows. *Fixed LRM* is the pipeline with a frozen latent reward model (Mi et al., 2026). *v2* and *w1p5* are co-evolving runs

started from the pretrained model with an absolute score target, differing only in the generator learning rate (2×10^{-6} and 5×10^{-6}); the first 200 steps of v2 update the reward head alone, so its step-200 checkpoint is the pretrained model. *dyn300* resumes v2 at step 300 on VisionReward motion pairs instead of VideoDPO, with and without an EMA generator. *oldw400ema*, *relgapema* and *kernelema* share one warm start: v2 at step 300 with the reward head re-fit for a further 100 steps. They then train the generator for 100 steps with the absolute target, the relative gap, and the relative gap plus kernel anchor respectively, and are evaluated with the EMA generator.

Two baselines bracket the results. The pretrained model sets the reference for IQ (0.679); the fixed reward leaves DD unchanged and lowers IQ by eight points. The naive co-evolving runs first raise DD (0.896 for v2 at step 300) and then collapse it (0.194 at step 750; 0.167 for w1p5 at step 450). The rise in SC, BC and MS along the way reflects videos that stop moving, not an improvement. Among the runs with the shared warm start, the absolute target continues the decline in DD (0.569). The relative gap holds DD at the pretrained level (0.694) with the best IQ of the block (0.659).

Table 8: Generator objective, evaluated at two checkpoints of the same 100-step run.

Objective	Step	Dyn.	Imaging	Aesth.	Subj.	Smooth
Absolute target	450	61.11	66.13	61.05	97.14	98.19
	500	56.94	64.09	61.24	96.74	98.28
Relative gap	450	72.22	64.96	60.47	96.25	98.14
	500	69.44	65.85	59.23	95.88	97.69