SCALING REINFORCEMENT LEARNING FOR DIFFUSION MODELS VIA VELOCITY MATCHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reward-based fine-tuning of diffusion models has largely inherited likelihood-based policy optimization developed for autoregressive large language models (LLMs). Diffusion models, however, are natively trained through velocity regression and do not directly provide the likelihood of a generated sample. Existing methods address this using transition likelihoods along stochastic denoising trajectories or evidence lower bounds (ELBOs) to approximate generated-sample likelihoods. These approximations arise from applying likelihood-based updates to models whose native training and generation operate through velocity fields. We instead take velocity matching as the starting point for reward fine-tuning. We propose **reward-based velocity matching (RVM)**, which weights the velocity-matching loss by reward with an additional anchor regression term, requiring neither likelihood estimation nor likelihood ratios. RVM recovers Reinforce Adjoint Matching at the update level and contains DiffusionNFT as a special case, while ELBO-based methods are closely related through the same velocity-regression structure. This unified view isolates reward design and anchor velocity as principal design choices in velocity-based fine-tuning. Across text-to-image, text-to-video, and image-to-video generation, RVM matches or outperforms the evaluated trajectory-based methods. On Wan2.1-T2V-1.3B, it achieves the highest VBench Overall among the evaluated methods, with substantially lower estimated training costs than the trajectory-based approaches. For video fine-tuning, we introduce a dynamic-tracking reward that provides explicit motion feedback and improves both Dynamic Degree and overall VBench performance.

Project page: https://rvm-submission.github.io/rvm/

1 INTRODUCTION

Reward-based fine-tuning for diffusion and flow models has become an important approach for adapting them to human preferences (Wu et al., 2023; Xu et al., 2023) and downstream criteria such as semantic alignment, visual quality, motion, and physical plausibility (Wang et al., 2026b; Tang et al., 2026; Xue et al., 2026b; Li et al., 2026b). A dominant approach transfers likelihood-based policy optimization developed for autoregressive large language models (LLMs), such as PPO and GRPO (Schulman et al., 2017; Shao et al., 2024), to diffusion models. For autoregressive LLMs, likelihood-based policy optimization is natural because sequence likelihoods factor into tractable next-token probabilities. However, diffusion models are fundamentally parameterized and trained through velocity matching, while the likelihood of a generated sample is not directly available.

Existing methods address this issue by reconstructing likelihood-based policy updates indirectly. *Trajectory-based* methods use tractable one-step transition probabilities $\pi_\theta(\boldsymbol{x}_{t-\Delta t} \mid \boldsymbol{x}_t)$ along stochastic denoising trajectories (Liu et al., 2025b; Xue et al., 2025). *ELBO-based* methods approximate the log-likelihood of a generated sample $\boldsymbol{x}_0$ using an evidence lower bound (ELBO) as a surrogate, computed from velocity-matching losses at independently noised states $\boldsymbol{x}_t$ (Xue et al., 2026a; Choi et al., 2026). Constructing $\boldsymbol{x}_t$ directly from the generated sample $\boldsymbol{x}_0$ decouples training updates from the rollout sampler, allowing fast deterministic ODE sampling. These ELBO-based policy-gradient updates can be expressed as weighted velocity matching, with weights given by the product of the reward and an estimated likelihood ratio. In practice, these likelihood-ratio estimates are biased and have high variance. More fundamentally, the need for such approximations arises

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.

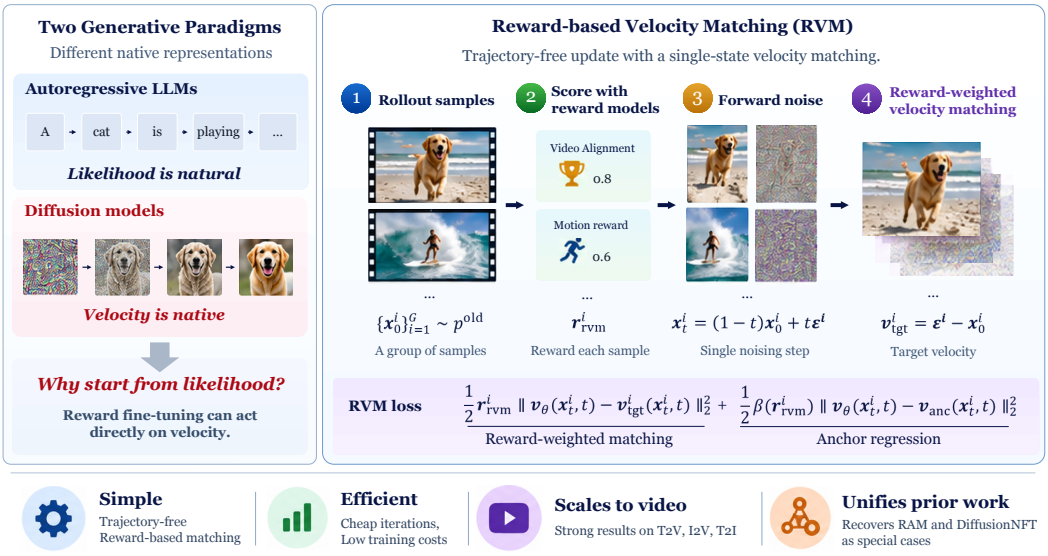


Figure 1: **Overview of reward-based velocity matching (RVM). Left:** Autoregressive LLMs operate naturally through token probabilities, whereas diffusion models are trained through velocity matching, motivating reward fine-tuning directly in velocity space. **Right:** RVM generates samples, scores them with reward models, and forward-noises each to an intermediate state. RVM objective combines reward-weighted velocity matching with anchor regression, requiring neither likelihood-ratio estimation nor stored rollout trajectories. This formulation supports different rollout samplers, applies to image and video generation, and recovers RAM and DiffusionNFT as special cases.

from applying likelihood-based policy optimization to models whose native training and generation operate through velocity fields. This mismatch raises our central question:

> *Can we formulate reward fine-tuning directly through velocity matching, without reconstructing the objective from likelihood-based policy optimization?*

We argue that reward fine-tuning can act directly on the native velocity representation of diffusion models, without reconstructing from likelihood estimation. We propose **reward-based velocity matching (RVM)**, an objective that weights the velocity-matching loss by reward with an additional anchor regression term. As illustrated in Fig. 1, given a generated sample $\boldsymbol{x}_0^i$, RVM independently construct noise state $\boldsymbol{x}_t^i$, and its velocity target $\boldsymbol{v}_{\text{tgt}}^i = \boldsymbol{\epsilon}^i - \boldsymbol{x}_0^i$, then weights its velocity-regression loss according to the sample reward r_{rvm}^i. Positive reward weights reinforce the velocity direction associated with the generated sample, while negative weights suppress it. The anchor term regularizes the predicted velocity toward a anchor field $\boldsymbol{v}_{\text{anc}}$, such as the velocity predicted by the frozen pretrained model. This penalizes excessive deviation from the reference model's denoising behavior during reward fine-tuning.

Recent methods such as RAM (Bergmeister et al., 2026) and DiffusionNFT (Zheng et al., 2026a) have demonstrated strong empirical performance in reward fine-tuning, motivated by stochastic control and contrastive learning, respectively. RVM unifies these methods, together with ELBO-based methods such as AWM (Xue et al., 2026a), within a weighted velocity matching framework. Specifically, RVM recovers their objectives at the level of updates under appropriate choices of sampling policy, reward weighting, and anchor, revealing a shared regression structure beneath their distinct motivations. This unified view organizes objective design around how rewards are defined and incorporated, and how the anchor field and its coefficient are chosen.

We evaluate the scalability and efficiency of RVM across text-to-image, text-to-video, and image-to-video generation using Stable Diffusion 3.5-Medium (Esser et al., 2024), Wan2.1-T2V-1.3B (Team Wan et al., 2025), and SkyReels-I2V (SkyReels-AI, 2025), respectively. RVM outperforms the evaluated trajectory-based methods in generation quality. Moreover, RVM substantially reduces both rollout and gradient-update costs relative to the evaluated trajectory-based methods. We also find that RVM achieves performance on par with or better than recent methods such as DiffusionNFT

and RAM, whose objectives can be recovered within our unified framework. We further analyze the effects of reward and anchor choices. Preference and visual-quality rewards can favor nearly static videos, whereas adding our dynamic-tracking (DT) reward improves motion and overall VBench performance. Overall, our **main contributions** are:

- We introduce **RVM**, which formulates reward fine-tuning directly in the native velocity representation of diffusion models. RVM combines reward-weighted velocity matching with an anchor regression term, requiring neither likelihood estimation nor likelihood ratios.
- Extensive experiments demonstrate that RVM scales across text-to-image, text-to-video, and image-to-video generation. Its trajectory-free updates and velocity matching loss enable efficient fine-tuning. RVM achieves competitive or superior performance across tasks, surpassing all evaluated trajectory- and velocity-based baselines in VBench Overall on Wan2.1-T2V-1.3B.
- We show that recent competitive methods, such as RAM, DiffusionNFT, and the ELBO-based approaches can be unified within the RVM framework. This unified view identifies reward design and anchor velocity as principal design choices within this family of objectives.
- We analyze the effects of reward design and anchor velocity. We introduce a dynamic-tracking reward that improves motion and overall video quality. The evaluated anchor configurations show different balances between overall quality and semantic alignment.

2 Background

2.1 Diffusion and Flow Models

Diffusion and flow-based models (Song et al., 2020; Lipman et al., 2023; Albergo et al., 2025; Liu et al., 2023) learn a data distribution on $\mathbb{R}^d$ by interpolating clean data $\boldsymbol{x}_0 \sim p_{\text{data}}$ and an independent noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ through $\boldsymbol{x}_t = (1-t)\boldsymbol{x}_0 + t\boldsymbol{\epsilon}$, or equivalently, $\boldsymbol{x}_t|\boldsymbol{x}_0 \sim \mathcal{N}((1-t)\boldsymbol{x}_0, t^2\boldsymbol{I})$. The model learns a velocity field $\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$ through a weighted regression:

$$\min_\theta \mathbb{E}_{t,\boldsymbol{x}_0,\boldsymbol{\epsilon}} \left[w(t) \|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v}_{\text{tgt}}\|^2 \right],$$

where $t \sim \text{Unif}(0,1)$, $\boldsymbol{v}_{\text{tgt}} := \boldsymbol{\epsilon} - \boldsymbol{x}_0$, and $w(t)$ is a positive weight function. To generate samples from the model, the standard practice is to simulate the ODE $\dot{\boldsymbol{x}}_t = \boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$ from $\boldsymbol{x}_1 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ to $\boldsymbol{x}_0$, or use the equivalent SDE samplers.

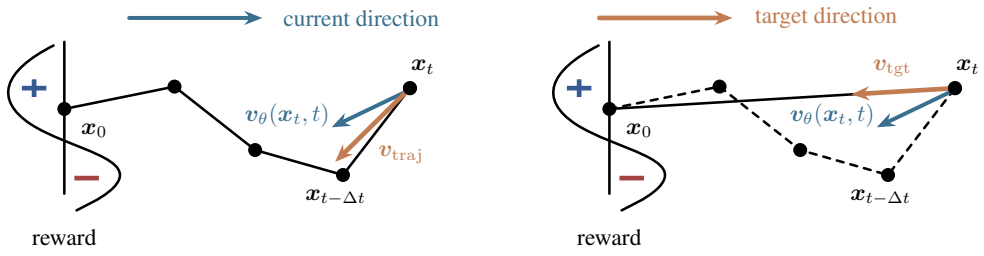
2.2 Reward Fine-tuning and Policy-Gradient Objectives

Reward fine-tuning via policy gradients Following RL fine-tuning of LLMs, reward fine-tuning treats a generative model as a policy π_θ and optimizes reward r through likelihood-based policy gradients such as REINFORCE (Sutton et al., 1999). Given samples from a sampling policy π_{old}, a standard importance-weighted policy-gradient objective can be written as

$$\min_\theta \; -\mathbb{E}_{\boldsymbol{x}\sim\pi_{\text{old}}} \left[r(\boldsymbol{x}) \bar{\rho}_\theta(\boldsymbol{x}) \log \pi_\theta(\boldsymbol{x}) \right] + \beta \, \text{KL}(\pi_\theta \| \pi_{\text{ref}}), \quad \bar{\rho}_\theta(\boldsymbol{x}) := \frac{\text{sg}[\pi_\theta(\boldsymbol{x})]}{\pi_{\text{old}}(\boldsymbol{x})}, \quad \text{(REINFORCE)}$$

where $\bar{\rho}_\theta$ is the detached importance ratio and β controls the regularization strength. The corresponding optimal policy satisfies $\pi^\star(\boldsymbol{x}) \propto \pi_{\text{ref}}(\boldsymbol{x}) \exp(r(\boldsymbol{x})/\beta)$. When conditioning on c, both the policy and reward depend on it, i.e., $\boldsymbol{x} \sim \pi_\theta(\cdot \mid c)$ and $r = r(\boldsymbol{x}, c)$; we omit c throughout for brevity. GRPO further uses group-relative rewards and clips the likelihood ratio. Given a group of G samples $\{\boldsymbol{x}^i\}_{i=1}^G$, its standardized reward is $r^i_{\text{grpo}} = \frac{r(\boldsymbol{x}^i) - \text{mean}_g\, r(\boldsymbol{x}^g)}{\text{std}_g\, r(\boldsymbol{x}^g)}$.

Trajectory-based approaches Because diffusion endpoint likelihoods, i.e., $\pi_\theta(\boldsymbol{x}_0)$, and their ratios are intractable, trajectory-based methods instead use tractable one-step transition probabilities $\pi_\theta(\boldsymbol{x}_{t-\Delta t} \mid \boldsymbol{x}_t)$ along stochastic denoising trajectories (Black et al., 2024; Fan et al., 2023). Flow-GRPO (Liu et al., 2025b) and DanceGRPO (Xue et al., 2025) apply GRPO-style updates to these transitions, requiring costly SDE rollouts and updating toward stochastic intermediate targets, which increases computational cost and can introduce additional variance.



(a) Trajectory-based. (b) Velocity-based.

Figure 2: We illustrate both updates under the same rollout and compare their loss targets. Given a generated endpoint $\boldsymbol{x}_0$ with *positive reward*, (a) the *trajectory-based* loss uses the target $\boldsymbol{v}_{\text{traj}}$ determined by the realized noisy transition from $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t-\Delta t}$, whereas (b) *velocity-based* losses, such as RVM, use the endpoint-defined target $\boldsymbol{v}_{\text{tgt}} = \boldsymbol{\epsilon} - \boldsymbol{x}_0$. Arrows indicate denoising directions. Negative rewards reverse the corresponding attraction into repulsion. See App. B.4 for details.

ELBO-based approaches Xue et al. (2026a); Choi et al. (2026) estimate the sample log-likelihood π_θ and importance ratio $\bar{\rho}_\theta$ in (REINFORCE) using a velocity-regression surrogate,

$$\log \pi_{\{\theta,\text{old}\}}(\boldsymbol{x}_0) \approx -\mathbb{E}_{t,\boldsymbol{\epsilon}}\left[w(t)\|\boldsymbol{v}_{\{\theta,\text{old}\}}(\boldsymbol{x}_t, t) - \boldsymbol{v}_{\text{tgt}}\|_2^2\right], \tag{ELBO}$$

where $\boldsymbol{x}_t = (1-t)\,\boldsymbol{x}_0 + t\,\boldsymbol{\epsilon}$ and $\boldsymbol{v}_{\text{tgt}} = \boldsymbol{\epsilon} - \boldsymbol{x}_0$. ELBO-based approaches estimate $\bar{\rho}_\theta$ by exponentiating the difference between the ELBO estimates for the current and sampling policies in (ELBO). Although effective in practice, this additional estimation step can make the resulting velocity updates sensitive to estimation noise. One variant is Advantage Weighted Matching (AWM) (Xue et al., 2026a), the on-policy version of this update, with $\pi_{\text{old}} := \pi_\theta$.

3 Reward Fine-Tuning in Velocity Space

3.1 Reward-Based Velocity Matching

The likelihood-based approaches above ask how to adapt likelihood-based policy gradients to diffusion models. We step back and ask whether likelihood should be the starting point at all. As illustrated in Fig. 1 (left), likelihood-based optimization naturally fits autoregressive LLMs, whose native outputs are token probabilities. Diffusion models, however, learn to generate through velocity fields, suggesting that reward feedback can act directly on this representation. This motivates **reward-based velocity matching (RVM)**: we formulate reward fine-tuning directly as velocity regression, removing the need to reconstruct likelihood-based policy updates.

Direct velocity-space objective We formulate reward fine-tuning directly in velocity space using two regression terms: reward-weighted velocity matching to incorporate reward feedback, and anchor regression to control model drift. As illustrated in Fig. 1 (right), the update follows four steps: generate samples $\boldsymbol{x}_0^i$ from sampling policy π_{old}, score them with reward models r_{rvm}^i, forward-noise them to states $\boldsymbol{x}_t^i = (1-t)\boldsymbol{x}_0^i + t\boldsymbol{\epsilon}^i$, where $\boldsymbol{\epsilon}^i \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, and perform reward-weighted velocity regression with targets $\boldsymbol{v}_{\text{tgt}}^i = \boldsymbol{\epsilon}^i - \boldsymbol{x}_0^i$ and anchor field $\boldsymbol{v}_{\text{anc}}$. These steps yield the following *reward-based velocity matching* (RVM) objective:

$$\mathcal{L}_{\text{rvm}} = \tfrac{1}{2}\,\mathbb{E}_{\boldsymbol{x}_0^{1:G}\sim\pi_{\text{old}}}\left[r_{\text{rvm}}^i\,\|\boldsymbol{v}_\theta(\boldsymbol{x}_t^i, t) - \boldsymbol{v}_{\text{tgt}}^i\|_2^2 + \beta(r_{\text{rvm}}^i)\,\|\boldsymbol{v}_\theta(\boldsymbol{x}_t^i, t) - \boldsymbol{v}_{\text{anc}}(\boldsymbol{x}_t^i, t)\|_2^2\right], \tag{RVM}$$

where $\beta : \mathbb{R} \to \mathbb{R}$ controls the strength of the anchor regularization. The first term incorporates reward feedback by weighting regression toward $\boldsymbol{v}_{\text{tgt}}^i$ with r_{rvm}^i: positive weights pull the predicted velocity toward the target, while negative weights push it away. The second term regularizes the prediction toward an *anchor field* $\boldsymbol{v}_{\text{anc}}$, such as the velocity predicted by a frozen pretrained model. When $\beta(r_{\text{rvm}}^i) > 0$, this term penalizes deviations from the anchor's denoising behavior and thereby controls model drift. As shown in Fig. 1 (right), both terms operate at the same independently forward-noised state, requiring neither likelihood-ratio estimation nor stored denoising trajectories. These *trajectory-free updates* decouple training from the rollout sampler, allowing samples to be generated using different samplers, including fast deterministic ODE solvers.

Reduced target variance and sampler-agnostic training Fig. 2 illustrates a source of high target variance in trajectory-based methods: their regression targets depend on the noisy intermediate state $\boldsymbol{x}_{t-\Delta t}$ sampled during an SDE rollout. Consequently, a positive reward reinforces the direction toward this intermediate sample, which can deviate from the direction toward the rewarded endpoint $\boldsymbol{x}_0$. RVM instead constructs its target directly as $\boldsymbol{v}_{\text{tgt}} = \boldsymbol{\epsilon} - \boldsymbol{x}_0 = (\boldsymbol{x}_t - \boldsymbol{x}_0)/t$ for $t > 0$, so the denoising direction $-\boldsymbol{v}_{\text{tgt}}$ points toward $\boldsymbol{x}_0$. This removes the target variability introduced by sampled reverse transitions (App. B.4). The same construction also makes loss evaluation independent of the rollout sampler: RVM requires only the generated endpoint and independently forward-noised states. It therefore supports both stochastic SDE and deterministic ODE rollouts without retaining intermediate trajectories, enabling flexible and memory-efficient reward fine-tuning.

3.2 A Unified View via Reward-based Velocity Matching

RAM (Bergmeister et al., 2026) and DiffusionNFT (Zheng et al., 2026a) have demonstrated strong empirical performance in reward fine-tuning. Although motivated by control-theoretic and contrastive-learning perspectives, respectively, both admit RVM representations under specific choices of sampling policy, reward, and anchor. Moreover, ELBO-based policy-gradient approaches such as (EPG) (Choi et al., 2026) and AWM (Xue et al., 2026a) share a common velocity matching structure.

RAM as an RVM update RAM regresses toward a detached reward-dependent target,

$$\mathcal{L}_{\text{ram}} = \tfrac{1}{2}\,\mathbb{E}_{\boldsymbol{x}_0^{1:G}\sim\pi_\theta}\left[\left\|\boldsymbol{v}_\theta(\boldsymbol{x}_t^i,t) - \text{sg}\left[\boldsymbol{v}_{\text{ref}}(\boldsymbol{x}_t^i,t) + r_{\text{ram}}^i\big(\boldsymbol{v}_{\text{tgt}}^i - \boldsymbol{v}_\theta(\boldsymbol{x}_t^i,t)\big)\right]\right\|_2^2\right], \tag{RAM}$$

whose gradient with respect to the predicted velicty $\boldsymbol{v}_\theta$ is [1]

$$\nabla_{\boldsymbol{v}_\theta}\mathcal{L}_{\text{ram}} = \boldsymbol{v}_\theta - \text{sg}\big[\boldsymbol{v}_{\text{ref}} + r_{\text{ram}}^i(\boldsymbol{v}_{\text{tgt}}^i - \boldsymbol{v}_\theta)\big] = r_{\text{ram}}^i\,(\boldsymbol{v}_\theta - \boldsymbol{v}_{\text{tgt}}^i) + (\boldsymbol{v}_\theta - \boldsymbol{v}_{\text{ref}}), \tag{1}$$

which exactly matches the RVM gradient $r^i(\boldsymbol{v}_\theta - \boldsymbol{v}_{\text{tgt}}^i) + \beta(\boldsymbol{v}_\theta - \boldsymbol{v}_{\text{anc}})$ with the reference anchor $\boldsymbol{v}_{\text{anc}} = \boldsymbol{v}_{\text{ref}}$ at unit strength $\beta = 1$ and reward weight $r_{\text{rvm}}^i = r_{\text{ram}}^i$. Thus, RAM is update-equivalent to on-policy RVM ($\pi_{\text{old}} = \pi_\theta$) with the frozen reference as its anchor.

DiffusionNFT as a special case of RVM The DiffusionNFT loss is[2]

$$\mathcal{L}_{\text{nft}} = \tfrac{1}{2\bar\beta}\mathbb{E}_{\boldsymbol{x}_0^{1:G}\sim\pi_{\text{old}}}\left[r_{\text{nft}}^i\|\boldsymbol{v}_\theta^+(\boldsymbol{x}_t^i,t) - \boldsymbol{v}_{\text{tgt}}^i\|_2^2 + (1 - r_{\text{nft}}^i)\|\boldsymbol{v}_\theta^-(\boldsymbol{x}_t^i,t) - \boldsymbol{v}_{\text{tgt}}^i\|_2^2\right]. \tag{NFT}$$

where $\boldsymbol{v}_\theta^\pm = \boldsymbol{v}_{\text{anc}} \pm \bar\beta(\boldsymbol{v}_\theta - \boldsymbol{v}_{\text{anc}})$ are the positive and negative velocity branches, $\boldsymbol{v}_{\text{anc}}$ is an EMA velocity, $\bar\beta > 0$ is the guidance strength, and $r_{\text{nft}}^i \in [0,1]$ is the reward weight. Expanding the squared norm yields

$$\mathcal{L}_{\text{nft}} = \tfrac{r_{\text{rvm}}^i}{2}\left\|\boldsymbol{v}_\theta - \boldsymbol{v}_{\text{tgt}}^i\right\|_2^2 + \tfrac{\bar\beta - r_{\text{rvm}}^i}{2}\left\|\boldsymbol{v}_\theta - \boldsymbol{v}_{\text{anc}}\right\|_2^2 + \text{const}, \qquad r_{\text{rvm}}^i := 2r_{\text{nft}}^i - 1. \tag{2}$$

Thus, (NFT) objective equals (RVM) up to constants for the EMA anchor and$\beta(r_{\text{rvm}}^i) = \bar\beta - r_{\text{rvm}}^i$.

Connection to ELBO-based policy gradients The ELBO-based approaches discussed above also admit a velocity-regression interpretation. Substituting the ELBO surrogate in (ELBO) into the policy-gradient formulation in (REINFORCE), together with reference regularization, gives the following surrogate for exact policy gradient (EPG) (Choi et al., 2026):

$$\mathcal{L}_{\text{epg}} = \mathbb{E}_{\boldsymbol{x}_0^{1:G}\sim\pi_{\text{old}}}\left[r_{\text{epg}}^i\bar\rho_\theta(\boldsymbol{x}_0^i)\|\boldsymbol{v}_\theta(\boldsymbol{x}_t^i,t) - \boldsymbol{v}_{\text{tgt}}^i\|^2 + \beta\|\boldsymbol{v}_\theta(\boldsymbol{x}_t^i,t) - \boldsymbol{v}_{\text{ref}}(\boldsymbol{x}_t^i,t)\|^2\right]. \tag{EPG}$$

Here, r_{epg}^i is the group-relative reward and $\bar\rho_\theta(\boldsymbol{x}_0^i)$ is a detached importance ratio estimated through the ELBO. This objective has the same weighted regression and anchor terms as RVM, but additionally weights the reward term by the estimated ratio.

In the on-policy case $\pi_{\text{old}} = \pi_\theta$, the importance ratio satisfies $\bar\rho_\theta = 1$, and (EPG) reduces to the AWM update (Xue et al., 2026a). RVM recovers this update, up to overall loss scaling, by setting $r_{\text{rvm}}^i = r_{\text{epg}}^i = r_{\text{awm}}^i$, $\boldsymbol{v}_{\text{anc}} = \boldsymbol{v}_{\text{ref}}$, and $\beta(r) \equiv \beta_{\text{awm}}$, where r_{awm}^i and β_{awm} denote the AWM advantage weight and constant reference-regularization coefficient, respectively. This connection

[1] We omit expectations in the gradient for simplicity.

[2] The factor $1/\bar\beta$ is a constant rescaling of the original DiffusionNFT (Zheng et al., 2026a) objective.

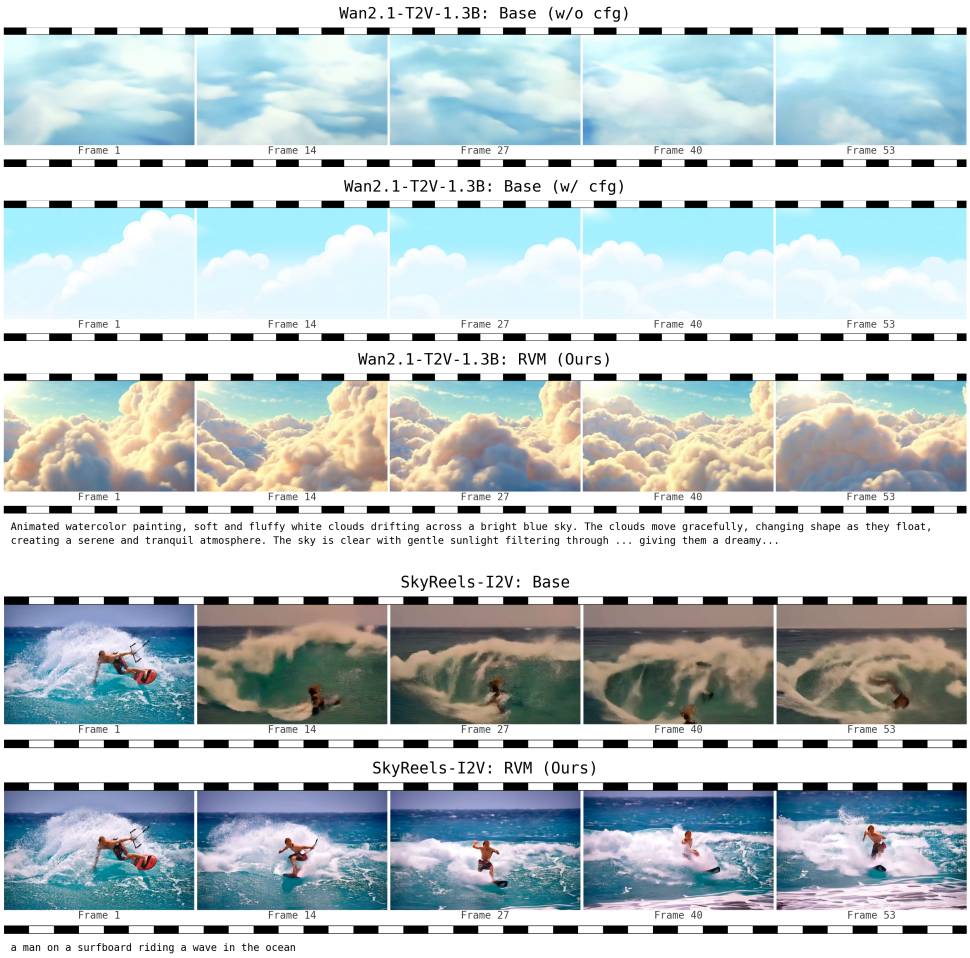


Figure 3: **Qualitative comparison between the base model and RVM** on Wan2.1-T2V-1.3B and SkyReels-I2V-V1. Additional examples are provided in App. G.3.

highlights the difference in starting point: RVM formulates these regression terms directly in velocity space, without reconstructing likelihood ratios.

The following theorem summarizes the equivalances introduced in this section:

Theorem 3.1 (Gradient-level Unification of Velocity-based Reward Objectives)**.** *Using the notation and objective conventions of Sec. 3.1, consider gradients at fixed training samples with reward weights and anchor velocities detached. The following equivalences hold.*

- ***RAM is update-equivalent to RVM.*** *With* $\pi_{\text{old}} = \pi_\theta$, $r^i_{\text{rvm}} = r^i_{\text{ram}}$, $\boldsymbol{v}_{\text{anc}} = \boldsymbol{v}_{\text{ref}}$, *and* $\beta \equiv 1$,
$$\nabla_{\boldsymbol{v}_\theta} \mathcal{L}_{\text{ram}} = \nabla_{\boldsymbol{v}_\theta} \mathcal{L}_{\text{rvm}}.$$
- ***DiffusionNFT is a special case of RVM.*** *With the EMA anchor,* $r^i_{\text{rvm}} = 2r^i_{\text{nft}} - 1$, *and* $\beta(r) = \bar{\beta} - r$,
$$\mathcal{L}_{\text{nft}} = \mathcal{L}_{\text{rvm}} + \text{const}, \qquad \nabla_{\boldsymbol{v}_\theta} \mathcal{L}_{\text{nft}} = \nabla_{\boldsymbol{v}_\theta} \mathcal{L}_{\text{rvm}},$$
where the additive constant is independent of θ.
- ***AWM is update-equivalent to RVM.*** *With* $\pi_{\text{old}} = \pi_\theta$, $r^i_{\text{rvm}} = r^i_{\text{awm}}$, $\boldsymbol{v}_{\text{anc}} = \boldsymbol{v}_{\text{ref}}$, *and constant* β,
$$\nabla_{\boldsymbol{v}_\theta} \mathcal{L}_{\text{awm}} = \nabla_{\boldsymbol{v}_\theta} \mathcal{L}_{\text{rvm}}.$$

See App. B.2 for the proof. Additionally, concurrent works are disucssed in (Xu et al., 2026; Anonymous, 2026) in App. A. This unified view identifies reward weighting and anchor velocity as principal design choices that govern how reward feedback reshapes the learned velocity field. We study their effects on generation behavior and quality in Sec. 4.3.

Table 1: **Evaluation results on Wan2.1-T2V-1.3B.** We report reward scores including VideoAlign, HPSv3 and VBench on VBench dataset. `Dyn. Degree` is the VBench dynamic-degree dimension, one of the seven quality dimensions already averaged into `Quality`, reported on its own because it is the only one that measures how much the video moves. Blue shading marks the first and second highest scores per column, including ties. †cited numbers from Li et al. (2026b).

	Reward				VBench			
Method	TA	VQ	MQ	HPSv3	Dyn. Degree	Quality	Semantic	Overall
Wan2.1-1.3B (w/o CFG)	2.53	2.33	0.35	1.85	54.17	78.59	65.73	76.02
Wan2.1-1.3B (w/ CFG)	5.50	4.08	0.97	7.85	65.28	83.72	80.64	83.10
Trajectory-based								
FlowGRPO	2.48	2.24	0.27	1.62	55.56	78.67	64.87	75.91
DanceGRPO†	−1.17	3.41	0.30	–	58.33	82.81	66.06	79.46
TaRoS†	−0.71	3.50	0.31	–	57.67	83.23	66.26	79.84
TaRoS-72B†	–	–	–	–	58.33	83.66	68.15	80.56
Velocity-based								
DiffusionNFT	5.03	5.04	1.63	9.70	63.89	84.37	77.28	82.95
RAM	4.86	4.23	1.14	8.39	61.11	83.35	76.33	81.95
Ours (RVM)	5.12	5.47	1.86	10.90	75.00	86.09	76.28	84.13

Table 2: **Evaluation results on SkyReels-I2V.** We report reward scores including VideoAlign, HPSv3 and VBench-I2V on VBench-I2V dataset. `Dyn. Degree` is the VBench dynamic-degree dimension, one of the seven quality dimensions already averaged into `Quality`. Blue shading marks the first and second highest scores per column, including ties. DiffusionNFT and RAM are trained with the same reward mixture as ours.

	Reward				VBench-I2V			
Method	TA	VQ	MQ	HPSv3	Dyn. Degree	I2V	Quality	Overall
SkyReels-I2V-V1 (w/o CFG)	2.45	2.30	0.06	3.18	64.63	87.79	75.57	81.68
SkyReels-I2V-V1 (w/ CFG)	2.53	1.99	0.21	4.56	47.97	88.87	75.36	82.12
Trajectory-based								
FlowGRPO	1.67	1.61	−0.47	−1.85	41.46	86.00	69.93	77.97
Velocity-based								
DiffusionNFT	2.65	2.59	0.61	6.43	47.56	93.18	79.15	86.16
RAM	2.68	2.56	0.49	6.57	51.22	93.49	79.72	86.61
Ours (RVM)	2.70	2.45	0.41	6.03	72.36	92.26	80.28	86.27

Table 3: **Evaluation results on SD3.5-M.** We report the OCR text-to-image fine-tuning task. Blue shading marks the first and second highest scores per column, including ties. †cited numbers from Choi et al. (2026). Note that OCR, PickScore, ClipScore, HPSv2.1, are used in training.

Method	OCR	PickScore	ClipScore	HPSv2.1	Aesthetic	ImgRwd
FlowGRPO†	0.92	22.41	0.290	0.280	5.32	0.95
AWM†	0.80	20.70	0.301	0.206	4.53	−0.13
DiffusionNFT†	0.93	22.09	0.307	0.277	5.17	0.97
PEPG†	0.94	22.93	0.315	0.302	5.33	1.34
Ours (RVM)	0.95	23.03	0.312	0.310	5.56	1.34

4 Experiments

We evaluate reward fine-tuning across text-to-image (T2I), text-to-video (T2V), and image-to-video (I2V) generation. The experiments address two aspects of RVM. In Sec. 4.2, we compare RVM with trajectory-based, ELBO-based, and other velocity-based methods, and quantify sampling and update costs. In Sec. 4.3, we examine how reward mixtures and anchor velocities affect video quality, motion, and semantic alignment. We compare generation with fewer inference steps in App. D.

4.1 Experimental Setup

Default configurations We fine-tune Stable Diffusion 3.5-Medium (Esser et al., 2024), Wan2.1-T2V-1.3B (Team Wan et al., 2025), and SkyReels-V1-I2V (SkyReels-AI, 2025) using LoRA (Hu et al., 2022) and deterministic ODE sampling. Unless otherwise stated, video training uses RVM without an anchor or CFG, with a weighted combination of VideoAlign text alignment and motion quality, HPSv3, and dynamic-tracking (DT) rewards. Full configurations including loss normalization and data protocols are provided in App. C. Details on video reward are elaborated in App. C.3.

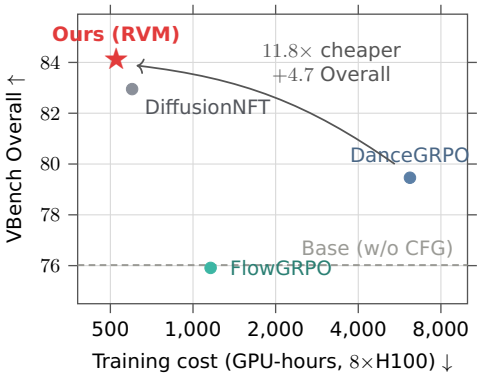


Figure 4: **Quality-cost comparison.** RVM achieves a 4.7-point higher VBench score at 11.8× lower training cost than DanceGRPO.

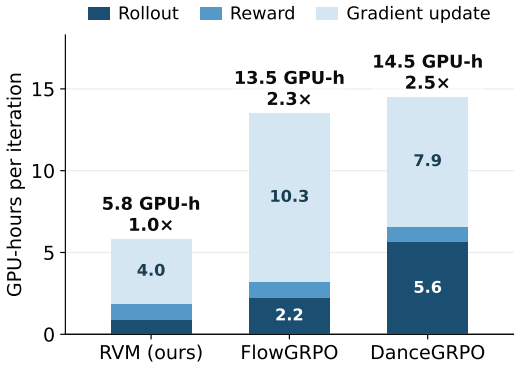


Figure 5: **Per-iteration GPU-hour cost on Wan2.1-1.3B.** Each iteration is decomposed into rollout, reward evaluation, and gradient update.

Evaluation We report reward-model scores alongside VBench (Huang et al., 2024) for T2V and VBench-I2V (Huang et al., 2026b) for I2V. T2I evaluation follows Choi et al. (2026) and reports OCR accuracy, image-quality metrics, and preference scores. Detailed protocols and complete video metrics are provided in Apps. C and G.2.

Baselines We compare RVM with pretrained models with and without CFG (scale 5 for Wan2.1 and 6 for SkyReels), trajectory-based FlowGRPO (Liu et al., 2025b), DanceGRPO (Xue et al., 2025), and TaRoS (Li et al., 2026b), and velocity-based DiffusionNFT (Zheng et al., 2026a) and RAM (Bergmeister et al., 2026). Published DanceGRPO and TaRoS results use their respective protocols with 50-step SDE sampling and CFG. T2I baselines additionally include ELBO-based AWM and PEPG, with baseline scores taken from Choi et al. (2026).

4.2 Main Results

RVM outperforms the trajectory-based methods Tabs. 1 to 3 compare RVM with trajectory-based fine-tuning across T2V, I2V, and the T2I OCR task. On Wan2.1-1.3B, RVM achieves 84.13 VBench Overall, outperforming FlowGRPO (75.91), DanceGRPO (79.46), and the TaRoS variants (79.84–80.56). On SkyReels, RVM reaches 86.27, exceeding FlowGRPO by 8.30 points. On the SD3.5-M OCR task, RVM improves OCR accuracy from FlowGRPO's 0.92 to 0.95 and surpasses it on all five accompanying metrics. These results demonstrate effective reward fine-tuning across image and video generation through direct velocity regression, without transition-likelihood updates.

Leading T2V and T2I performance On Wan2.1-T2V-1.3B (Tab. 1), RVM achieves the highest scores on six of eight reported metrics across all methods and pretrained baselines, including a VBench Overall score of 84.13, surpassing all evaluated trajectory- and velocity-based baselines. It exceeds DiffusionNFT and RAM by 1.18 and 2.18 Overall points, respectively, while attaining the highest Quality and Dynamic Degree. On SkyReels (Tab. 2), RVM also leads in Quality and Dynamic Degree, with Overall performance comparable to DiffusionNFT and RAM. On SD3.5-M OCR (Tab. 3), RVM achieves the best or tied-best results on five of six metrics and outperforms AWM and DiffusionNFT on all six. These results establish direct velocity regression as a strong approach across video and image generation, without likelihood-ratio estimation. We examine reward and anchor choices in Sec. 4.3.

Lower training costs than trajectory-based methods Fig. 4 compares generation quality against total training cost on Wan2.1. RVM achieves a 4.7-point higher VBench Overall score than DanceGRPO at 11.8× lower total cost, using approximately 525 GPU-hours over 90 iterations. The DanceGRPO cost in Fig. 4 is estimated by extrapolating measured rollout, reward, and update timings to 220 training iterations, using 512 videos per iteration (32 prompts with group size 16), yielding approximately 6,171 GPU-hours. To explain these savings, Fig. 5 breaks down the estimated per-iteration costs at a common batch size of 256 videos. RVM requires 5.8 GPU-hours, compared with 13.5 for FlowGRPO and 14.5 for DanceGRPO, corresponding to approximately 2.3× and 2.5× lower costs. Both sampling and updates contribute: our deterministic ODE rollouts without CFG

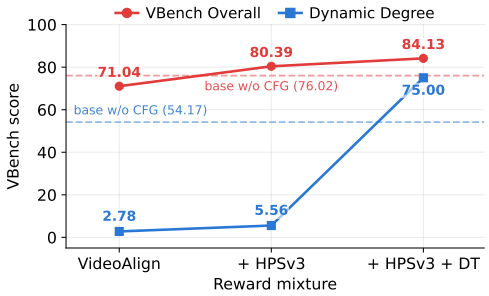


Figure 6: **Ablation of reward design on Wan2.1-T2V-1.3B.** VBench Overall and dynamic degree for the three reward mixtures.

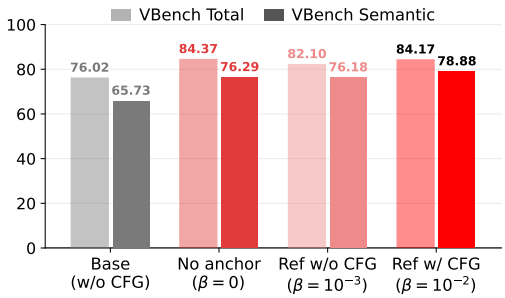


Figure 7: **Ablation of the anchor velocity on Wan2.1-T2V-1.3B.**

require fewer model evaluations, while velocity-regression updates use fewer noise levels than the compared trajectory-based configurations. These estimates use measured component timings and the respective training configurations, comparing the costs and quality of the reported runs rather than time to a common quality threshold.

4.3 Further Analysis

Reward design Reward design determines which generation behaviors are reinforced during fine-tuning, making it essential to capture the properties we ultimately want to improve. For video generation, this requires rewarding both visual quality and motion. Common choices include **(i) VideoAlign** (Liu et al., 2025c), which provides rewards for text alignment (TA), motion quality (MQ), and visual quality (VQ). For **(ii) VideoAlign + HPSv3**, we incorporate human-preference feedback using HPSv3 (Ma et al., 2025). In our experiments, however, fine-tuning with these reward mixtures substantially reduces motion, even when a motion-quality reward is included. This motivates our **dynamic-tracking (DT) reward**, which uses RAFT optical flow (Teed & Deng, 2020) to explicitly reward motion dynamics. To examine the contributions of preference and explicit motion feedback, we compare **(i)** and **(ii)** with **(iii) VideoAlign + HPSv3 + DT**. Fig. 6 reveals that improved aggregate scores can conceal a substantial loss of motion. VideoAlign alone yields a Dynamic Degree of just 2.78, compared with 54.17 for the pretrained model without CFG. Replacing VQ with HPSv3 raises VBench Overall from 71.04 to 80.39, yet leaves Dynamic Degree at only 5.56. Stronger preference feedback improves the overall score without recovering motion. Adding DT changes this outcome, raising Dynamic Degree to 75.00 while further improving Overall to 84.13, surpassing the pretrained baseline on both measures. Thus, explicit motion feedback enables simultaneous gains in motion and aggregate performance in this setting. We use the third mixture for video training; weights and model-specific reward definitions are provided in App. C.3.

Anchor velocity We compare training without an anchor with training anchored to the frozen base-model velocity evaluated either without CFG or with CFG where $\beta > 0$. Fig. 7 shows that the model trained with CFG-guided anchor achieves a Semantic score of 78.88, compared with 76.18 for the model trained with the reference anchor without CFG and 76.29 for the model trained without an anchor. The positive anchor term in (RVM) penalizes deviations from the guided reference field during training, while inference uses the learned velocity without CFG. The unanchored setting retains the highest VBench Total, reaching 84.37 compared with 84.17 for the CFG-guided anchor.

5 Conclusion

We take velocity matching as the starting point for reward fine-tuning, yielding RVM and a common regression structure connecting RAM and DiffusionNFT. Across image and video benchmarks, RVM achieves performance on par with or better than the evaluated baselines, with lower estimated training costs than trajectory-based methods on Wan2.1-T2V-1.3B. Exploring better reward designs and anchor choices is an interesting direction for future work. Extending RVM to autoregressive video generation would also be a promising direction.

AI USE STATEMENT

We used generative AI tools to improve the readability of the manuscript, including polishing sentences and correcting grammar. The authors reviewed all AI-assisted edits and take full responsibility for the content of this work.

REPRODUCIBILITY STATEMENT

App. B provides the loss definitions, reward-weight conventions, and derivations connecting the objectives. App. C.1 describes optimization, sampling, loss normalization, training and evaluation data, and the anchor evaluation protocol; Tab. 5 lists the per-model hyperparameters and hardware. App. C.2 details the text-to-image OCR task, including its training rewards and evaluation settings, while App. C.3 defines the video rewards and their mixture weights. App. D specifies the few-step inference schedules and evaluation procedure. Full per-dimension video benchmark results and additional qualitative comparisons are provided in App. G.2 and App. G.3, respectively.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. pp. 12248–12267, August 2024. doi: 10.18653/v1/2024.acl-long.662. URL https://aclanthology.org/2024.acl-long.662/.

Michael Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025. URL http://jmlr.org/papers/v26/23-1605.html.

Anonymous. FlowSRL: Stable and Scalable Reinforcement Learning for Diffusion and Flow Models. Anonymous submission to ICLR 2027, 2026.

Andreas Bergmeister, Stefanie Jegelka, Nikolas Nüsken, Carles Domingo-Enrich, and Jakiw Pidstrigach. Reinforce adjoint matching: Scaling RL post-training of diffusion and flow-matching models, 2026. URL https://arxiv.org/abs/2605.10759.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. 2024. URL https://openreview.net/forum?id=YCWjhGrJFD.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. MiniMax-M1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

Huayu Chen, Kaiwen Zheng, Qinsheng Zhang, Ganqu Cui, Yin Cui, Haotian Ye, Tsung-Yi Lin, Ming-Yu Liu, Jun Zhu, and Haoxiang Wang. NFT: Bridging supervised learning and reinforcement learning in math reasoning. 2026. URL https://openreview.net/forum?id=ujBrsQm6Zu.

Jaemoo Choi, Yuchen Zhu, Wei Guo, Petr Molodyk, Bo Yuan, Jinbin Bai, Yi Xin, Molei Tao, and Yongxin Chen. Rethinking the design space of reinforcement learning for diffusion models: On the importance of likelihood estimation beyond loss design. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=3mJ9eTU319.

Carles Domingo-Enrich, Michal Drozdzal, Brian Karrer, and Ricky T. Q. Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xQBRrtQM8u.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=FPnUhsQJ5B.

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. 2023. URL https://openreview.net/forum?id=8OTPepXzeh.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dailan He, Guanlin Feng, Xingtong Ge, Yi Zhang, Bingqi Ma, Guanglu Song, Yu Liu, and Hongsheng Li. AR-CoPO: Align autoregressive video generation with contrastive policy optimization, 2026a. URL https://arxiv.org/abs/2603.17461.

Xiaoxuan He, Siming Fu, Yuke Zhao, Wanli Li, Jian Yang, Dacheng Yin, Fengyun Rao, and Bo Zhang. TempFlow-GRPO: When timing matters for GRPO in flow models. 2026b. URL https://openreview.net/forum?id=7mCo3R3Wyn.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7514–7528, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.595. URL https://aclanthology.org/2021.emnlp-main.595/.

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. 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Runhui Huang, Jie Wu, Rui Yang, Zhe Liu, and Hengshuang Zhao. AlphaGRPO: Unlocking self-reflective multimodal generation in UMMs via decompositional verifiable reward, 2026a. URL https://arxiv.org/abs/2605.12495.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, June 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):3268–3285, 2026b. doi: 10.1109/TPAMI.2025.3633890.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

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. 2023. URL https://openreview.net/forum?id=G5RwHpBUv0.

Black Forest Labs. Flux. https://github.com/black-forest-labs/flux, 2024.

Doyeon Lee, Eunyi Lyou, Hyunsoo Cho, Soo Kyung Kim, Joonseok Lee, and Jaemoo Choi. QUATRO: Query-adaptive trust region policy optimization for LLM fine-tuning. 2026a. URL https://openreview.net/forum?id=98XhDbtB21.

Jeongjae Lee, Jinho Chang, Jeongsol Kim, and Jong Chul Ye. Reward score matching: Unifying reward-based fine-tuning for flow and diffusion models, 2026b. URL https://arxiv.org/abs/2604.17415.

Junzhe Li, Yutao Cui, Tao Huang, Yinping Ma, Chun Fan, Yiming Cheng, Miles Yang, Zhao Zhong, and Liefeng Bo. MixGRPO: Unlocking flow-based GRPO efficiency with mixed ODE-SDE, 2026a. URL https://arxiv.org/abs/2507.21802.

Rui Li, Yuanzhi Liang, Ziqi Ni, Haibing Huang, Chi Zhang, and Xuelong Li. Rethinking reward signals in video GRPO: When scores become targets, 2026b. URL https://arxiv.org/abs/2511.19356.

Yuming Li, Yikai Wang, Yuying Zhu, Zhongyu Zhao, Ming Lu, Qi She, and Shanghang Zhang. BranchGRPO: Stable and efficient GRPO with structured branching in diffusion models. In *The Fourteenth International Conference on Learning Representations*, 2026c. URL https://openreview.net/forum?id=T2nP2IQasd.

Zhanhao Liang, Tao Yang, Jie Wu, Chengjian Feng, and Liang Zheng. LeapAlign: Post-training flow matching models at any generation step by building two-step trajectories, 2026. URL https://arxiv.org/abs/2604.15311.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Zhuo Jiang. When speed kills stability: Demystifying RL collapse from the training-inference mismatch, September 2025a. URL https://richardli.xyz/rl-collapse.

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. 2025b. URL https://openreview.net/forum?id=oCBKGw5HNf.

Jie Liu, Gongye Liu, Jiajun Liang, Ziyang Yuan, Xiaokun Liu, Mingwu Zheng, Xiele Wu, Qiulin Wang, Menghan Xia, Xintao Wang, Xiaohong Liu, Fei Yang, Pengfei Wan, Di Zhang, Kun Gai, Yujiu Yang, and Wanli Ouyang. Improving video generation with human feedback. 2025c. URL https://openreview.net/forum?id=nHkg4yc7SP.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025d.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2206.00927.

Yanzuo Lu, Ronglai Zuo, and Jiankang Deng. RAVEN: Real-time autoregressive video extrapolation with consistency-model GRPO, 2026. URL https://arxiv.org/abs/2605.15190.

Yuhang Ma, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards wide-spectrum human preference score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15086–15095, October 2025.

Weijia Mao, Hao Chen, Zhenheng Yang, and Mike Zheng Shou. The image as its own reward: Reinforcement learning with adversarial reward for image generation, 2025. URL https://arxiv.org/abs/2511.20256.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. 35:27730–27744, 2022. doi: 10.52202/068431-2011. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Found. Trends Optim.*, 1(3):127239, January 2014. ISSN 2167-3888. doi: 10.1561/2400000003. URL https://doi.org/10.1561/2400000003.

Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via Q-score matching. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=35ahHydjXo.

Allen Z. Ren, Justin Lidard, Lars Lien Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mEpqHvbD2h.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

SkyReels-AI. SkyReels V1: Human-centric video foundation model. https://github.com/SkyworkAI/SkyReels-V1, 2025.

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.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Bingda Tang, Yuhui Zhang, Xiaohan Wang, Jiayuan Mao, Ludwig Schmidt, and Serena Yeung-Levy. V-GRPO: Online reinforcement learning for denoising generative models is easier than you think, 2026. URL https://arxiv.org/abs/2604.23380.

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.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020.

Fu-Yun Wang, Han Zhang, Michaël Gharbi, Hongsheng Li, and Taesung Park. PromptRL: Prompt matters in RL for flow-based image generation. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=YKamMflrv7.

Jin Wang, Jianxiang Lu, Guangzheng Xu, Comi Chen, Haoyu Yang, Linqing Wang, Peng Chen, Mingtao Chen, Zhichao Hu, Longhuang Wu, Shuai Shao, Qinglin Lu, and Ping Luo. TAGRPO: Boosting GRPO on image-to-video generation with direct trajectory alignment. In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=NEiMxuUCSc.

Jing Wang, Jiajun Liang, Jie Liu, Henglin Liu, Gongye Liu, Jun Zheng, Wanyuan Pang, Ao Ma, Zhenyu Xie, Xintao Wang, et al. GRPO-Guard: Mitigating implicit over-optimization in flow matching via regulated clipping. *arXiv preprint arXiv:2510.22319*, 2025a.

Yibin Wang, Zhimin Li, Yuhang Zang, Yujie Zhou, Jiazi Bu, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Pref-GRPO: Pairwise preference reward-based GRPO for stable text-to-image reinforcement learning. *arXiv preprint arXiv:2508.20751*, 2025b.

Yifan Wang, Yanyu Li, Gordon Guocheng Qian, Sergey Tulyakov, Yun Fu, and Anil Kag. Diffusion-DRF: Free, rich, and differentiable reward for video diffusion fine-tuning, 2026c. URL https://arxiv.org/abs/2601.04153.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2096–2105, October 2023.

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. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=JVzeOYEx6d.

Yixian Xu, Yuanrui Zhang, Shengjie Luo, Liwei Wang, and Di He. Designing reinforcement learning for diffusion models: A unified path-space view, 2026. URL https://arxiv.org/abs/2608.14430.

Shuchen Xue, Chongjian GE, Shilong Zhang, Yichen Li, and Zhi-Ming Ma. Advantage weighted matching: Aligning RL with pretraining in diffusion models. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=nLY2pOYBrJ.

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.

Zeyue Xue, Siming Fu, Jie Huang, Shuai Lu, Haoran Li, Yijun Liu, Yuming Li, Xiaoxuan He, Mengzhao Chen, Haoyang Huang, Nan Duan, and Ping Luo. A systematic post-train framework for video generation, 2026b. URL https://arxiv.org/abs/2604.25427.

Haotian Ye, Kaiwen Zheng, Jiashu Xu, Puheng Li, Huayu Chen, Jiaqi Han, Sheng Liu, Qinsheng Zhang, Hanzi Mao, Zekun Hao, et al. Data-regularized reinforcement learning for diffusion models at scale. *arXiv preprint arXiv:2512.04332*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Shenghai Yuan, Jinfa Huang, Xianyi He, Yunyang Ge, Yujun Shi, Liuhan Chen, Jiebo Luo, and Li Yuan. Identity-preserving text-to-video generation by frequency decomposition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=Loek7hfb46P.

Qinsheng Zhang, Molei Tao, and Yongxin Chen. gDDIM: Generalized denoising diffusion implicit models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1hKE9qjvz-.

Ruicheng Zhang, Kaixi Cong, Jun Zhou, Zhizhou Zhong, Zunnan Xu, Shuiyang Mao, Wei Liu, and Xiu Li. KVPO: ODE-native GRPO for autoregressive video alignment via KV semantic exploration, 2026a. URL https://arxiv.org/abs/2605.14278.

Yifan Zhang, Yifeng Liu, Huizhuo Yuan, Yang Yuan, Quanquan Gu, and Andrew C Yao. On the design of KL-regularized policy gradient algorithms for LLM reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=qe060gmfm7.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

Chujie Zheng, Kai Dang, Bowen Yu, Mingze Li, Huiqiang Jiang, Junrong Lin, Yuqiong Liu, Hao Lin, Chencan Wu, Feng Hu, et al. Stabilizing reinforcement learning with LLMs: Formulation and practices. *arXiv preprint arXiv:2512.01374*, 2025a.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025b.

Kaiwen Zheng, Huayu Chen, Haotian Ye, Haoxiang Wang, Qinsheng Zhang, Kai Jiang, Hang Su, Stefano Ermon, Jun Zhu, and Ming-Yu Liu. DiffusionNFT: Online diffusion reinforcement with forward process. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=VJZ477R89F.

Mingzhe Zheng, Weijie Kong, Yue Wu, Dengyang Jiang, Yue Ma, Xuanhua He, Bin Lin, Kaixiong Gong, Zhao Zhong, Liefeng Bo, Qifeng Chen, and Harry Yang. Manifold-aware exploration for reinforcement learning in video generation, 2026b. URL https://arxiv.org/abs/2603.21872.

A Related Works

RL fine-tuning for language models RL has been extensively studied as a principled approach for improving language-model capabilities and alignment (Ouyang et al., 2022; Guo et al., 2025). Popular methods include PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), and their variants (Liu et al., 2025d; Yu et al., 2025; Ahmadian et al., 2024). Recent algorithmic developments focus on designing more stable policy objectives (Zhao et al., 2025; Zheng et al., 2025b; Gao et al., 2025; Chen et al., 2025; Kimi Team et al., 2025) and understanding their estimation and optimization behavior (Zhang et al., 2026b; Zheng et al., 2025a; Liu et al., 2025a). QUATRO (Lee et al., 2026a) is closely related to our policy-gradient formulation, as it studies proximal sequence-level policy optimization for likelihood-based LLMs. However, directly transferring these methods to diffusion and flow models is nontrivial because diffusion-model likelihoods and likelihood ratios are intractable.

Trajectory-based RL for diffusion and flow models RL has also been widely adopted to fine-tune diffusion and flow models toward human preferences or task-specific rewards (Fan et al., 2023; Black et al., 2024; Domingo-Enrich et al., 2025). One line of work follows a trajectory-based view, where the denoising or flow trajectory is treated as the policy trajectory. FlowGRPO (Liu et al., 2025b) adapts GRPO to diffusion models using trajectory-based likelihood estimation for text-to-image reward fine-tuning. Subsequent works modify the trajectory estimators, samplers, or rollout structures (He et al., 2026b; Wang et al., 2025a; Xue et al., 2025; Wang et al., 2025b; Ye et al., 2025; Xue et al., 2026a). Other examples include MixGRPO (Li et al., 2026a), BranchGRPO (Li et al., 2026c), and LeapAlign (Liang et al., 2026), which modify the rollout or trajectory construction. Reward Score Matching (Lee et al., 2026b) unifies many of these trajectory-based methods as score-matching regressions toward a value-guided target, and can be viewed as a trajectory-level counterpart of the ELBO-based approaches (EPG). For FlowGRPO and DanceGRPO, we quantify rollout and gradient-update costs under the Wan2.1 configurations evaluated in Sec. 4.2.

Simplified and trajectory-independent diffusion fine-tuning A separate line of work seeks simpler alternatives that avoid explicit trajectory-level likelihood estimation. DiffusionNFT (Zheng et al., 2026a) adapts negative fine-tuning (Chen et al., 2026) to diffusion and flow models by directly modifying velocity-regression losses. AWM (Xue et al., 2026a) and RAM (Bergmeister et al., 2026) similarly replace trajectory-based policy-gradient estimation with simpler velocity- or score-based objectives. Relatedly, DPPO (Ren et al., 2025) applies PPO-style updates to diffusion policies in continuous-control and robotic-manipulation settings, while Q-score matching (Psenka et al., 2024) learns diffusion policies from reward signals by matching score functions with Q-function gradients. Choi et al. (2026) show that, for diffusion fine-tuning, simple objectives can perform competitively when the likelihood or velocity-based training loss is chosen carefully. Our work follows this direction, focusing on trajectory-independent velocity-based losses and analyzing when objective engineering is actually necessary. Concurrently, Xu et al. (2026) present a unified path-space view in which reverse-trajectory and forward-matching methods arise from a common importance-sampling-based policy-gradient principle, attributing empirical gaps between method families to variance reduction rather than differing RL principles. Their unification operates on trajectory-space likelihood ratios, whereas RVM unifies velocity-based methods directly in the model's native velocity representation, avoiding likelihood estimation altogether. Concurrent work by Anonymous (2026) develops variants of velocity-matching RL objectives. Complementing our focus on objective unification and reward design, this work explores combinations with search algorithms and flow maps to scale to challenging rewards.

RL fine-tuning for video diffusion models Recent works have begun to apply RL-style fine-tuning to video diffusion and autoregressive video generators. DanceGRPO (Xue et al., 2025) and FlowGRPO-based video extensions apply GRPO-style objectives to video generation, while KVPO (Zhang et al., 2026a), RAVEN (Lu et al., 2026), TAGRPO (Wang et al., 2026b), SAGE-GRPO (Zheng et al., 2026b), AR-CoPO (He et al., 2026a), V-GRPO (Tang et al., 2026), and recent systematic video fine-tuning frameworks (Xue et al., 2026b) explore different objectives, rollout strategies, and video-specific reward designs. Other works focus on reward construction, including TaRoS (Li et al., 2026b), Adv-GRPO (Mao et al., 2025), Diffusion-DRF (Wang et al., 2026c), AlphaGRPO (Huang et al., 2026a), and PromptRL (Wang et al., 2026a). With the evaluated reward mixtures, RVM exceeds

FlowGRPO in VBench Overall on Wan2.1 and VBench-I2V Overall on SkyReels, while requiring lower estimated training cost than FlowGRPO and DanceGRPO on Wan2.1 (Sec. 4.2).

B Discussion of Loss Functionals

We collect here the loss functionals discussed in Sec. 3.1 in a common notation and describe the reward weight and the per-loss implementation choices used in our runs. Throughout, $\boldsymbol{x}_0^{1:G}$ is a group of G samples drawn for a shared prompt from each method's sampling policy, a general sampling π_{old} for RVM, the current policy π_θ for RAM, and its exponential moving average π_{old} for DiffusionNFT, while $\boldsymbol{x}_t^i \sim p_{t|0}(\cdot \mid \boldsymbol{x}_0^i)$ is a single noisy state with velocity target $\boldsymbol{v}_{\text{tgt}}^i$, and r^i is the reward weight of the i-th sample.

Reward weight For video training, we center rewards within each prompt group and normalize by the standard deviation across the full batch. The implemented standardized reward is

$$r_{\text{grpo}}^i = \frac{R(\boldsymbol{x}_0^i) - \text{mean}\left(R(\boldsymbol{x}_0^1), \ldots, R(\boldsymbol{x}_0^G)\right)}{\text{std}_{\text{batch}}(R)},$$

with reward weights $r_{\text{rvm}}^i = r_{\text{ram}}^i = r_{\text{grpo}}^i$ for the video runs. OCR instead uses prompt-group mean subtraction without standard-deviation normalization (App. C.2). These settings are task- and method-specific, rather than shared by all baselines. DiffusionNFT maps the unscaled standardized reward to a convex mixing weight $r_{\text{nft}}^i \in [0, 1]$, as detailed below.

B.1 Velocity-based Loss Functionals

RVM The objective in (RVM) is the velocity-regression form we optimize, a reward-weighted velocity regression $r_{\text{rvm}}^i \, \|\boldsymbol{v}_\theta(\boldsymbol{x}_t^i, t) - \boldsymbol{v}_{\text{tgt}}^i\|_2^2$ where the reward acts as a signed coefficient on the endpoint-defined velocity target. Positive rewards encourage the model to match the velocity direction associated with the generated sample, whereas negative rewards discourage that direction. It uses the task-specific reward weight described above and the ELBO weighting $w(t)$ of (ELBO) (App. B.3), and forms no likelihood ratio and simulates no trajectory.

RAM RAM (RAM) dispenses with the likelihood ratio entirely and instead shapes the velocity-regression target. In our runs the reward weight r_{ram}^i in (RAM) is the prompt-group-centered, batch-normalized reward above; it multiplies the residual $\boldsymbol{v}_{\text{tgt}}^i - \boldsymbol{v}_\theta(\boldsymbol{x}_t^i, t)$ between the bridge target and the current velocity, nudging $\boldsymbol{v}_\theta$ away from the frozen reference $\boldsymbol{v}_{\text{ref}}$ in proportion to r_{ram}^i. The original formulation instead scales this residual by the raw scalar reward, with no group baseline (Bergmeister et al., 2026). In either case the reward weight acts as a multiplicative scale on the regression target rather than as a weight on a log-probability, and the reference enters as an additive anchor inside the stop-gradient target instead of as a separate $\beta \, \|\boldsymbol{v}_\theta - \boldsymbol{v}_{\text{ref}}\|_2^2$ regularizer.

DiffusionNFT DiffusionNFT (NFT) also avoids log-probabilities, contrasting the two implicit velocity fields $v_\theta^\pm = (1 \mp \bar{\beta})\boldsymbol{v}_{\text{anc}} \pm \bar{\beta}\boldsymbol{v}_\theta$, where $\boldsymbol{v}_{\text{anc}}$ is an EMA-updated velocity (an exponential moving average of the current policy) and $\bar{\beta}$ is the guidance strength; we use $\bar{\beta} = 0.1$. Its reward weight is a *squashed* reward $r_{\text{nft}}^i \in [0, 1]$ that acts as a convex mixing weight between a positive branch, pulled toward the bridge target $\boldsymbol{v}_{\text{tgt}}^i$, and a negative branch, pushed away from it. Because the weights r_{nft}^i and $1 - r_{\text{nft}}^i$ must form a convex combination, the standardized reward cannot be used directly; we map it into $[0, 1]$ by clipping and rescaling. Concretely, r_{nft}^i is obtained from the unscaled, batch-normalized reward r_{grpo}^i defined above by

$$r_{\text{nft}}^i = \text{clip}\big(\tfrac{1}{2} + \tfrac{1}{10}\ \text{clip}(r_{\text{grpo}}^i, -5, 5),\ 0,\ 1\big)\,,$$

so that a group-average sample ($r_{\text{grpo}}^i = 0$) maps to $\frac{1}{2}$, while $r_{\text{grpo}}^i = +5$ and $r_{\text{grpo}}^i = -5$ map to 1 and 0, respectively (the inner clip range ± 5 is the clipping constant used here). High-reward samples ($r_{\text{nft}}^i \to 1$) place more weight on the positive regression branch, while low-reward samples ($r_{\text{nft}}^i \to 0$) place more weight on the negative regression branch. For DiffusionNFT-specific implementation details, including EMA decay and update scheduling, loss normalization, timestep sampling, and training budget, see Zheng et al. (2026a).

B.2 Connections between the Loss Functionals

Proof of the unification RAM and DiffusionNFT are motivated by different principles: RAM is derived from adjoint matching (Bergmeister et al., 2026), whereas DiffusionNFT is derived from contrastive flow matching (Zheng et al., 2026a). Nevertheless, their velocity-field updates share the same anchored regression form as (RVM). Specifically, RAM corresponds to choosing the frozen reference velocity as the anchor with unit anchor strength, while DiffusionNFT corresponds to choosing the EMA velocity as the anchor with reward-dependent anchor strength $\bar{\beta} - r^i_{\mathrm{rvm}}$.

Proof of Thm. 3.1. Omit the shared argument $(\boldsymbol{x}^i_t, t)$ on every velocity and recall the RVM gradient $\nabla_{\boldsymbol{v}_\theta}\mathcal{L}_{\mathrm{rvm}} = r^i(\boldsymbol{v}_\theta - \boldsymbol{v}^i_{\mathrm{tgt}}) + \beta(\boldsymbol{v}_\theta - \boldsymbol{v}_{\mathrm{anc}})$.

(i) RAM. The target of (RAM) is detached, so the squared-error gradient is current minus target,

$$\nabla_{\boldsymbol{v}_\theta}\mathcal{L}_{\mathrm{ram}} = \boldsymbol{v}_\theta - \mathrm{sg}\big[\boldsymbol{v}_{\mathrm{ref}} + r^i_{\mathrm{ram}}(\boldsymbol{v}^i_{\mathrm{tgt}} - \boldsymbol{v}_\theta)\big] = r^i_{\mathrm{ram}}\,(\boldsymbol{v}_\theta - \boldsymbol{v}^i_{\mathrm{tgt}}) + (\boldsymbol{v}_\theta - \boldsymbol{v}_{\mathrm{ref}}), \tag{3}$$

which is $\nabla_{\boldsymbol{v}_\theta}\mathcal{L}_{\mathrm{rvm}}$ at $\boldsymbol{v}_{\mathrm{anc}} = \boldsymbol{v}_{\mathrm{ref}}$, $\beta = 1$, $r^i = r^i_{\mathrm{ram}}$. The two gradients agree at every $\boldsymbol{v}_\theta$.

(ii) DiffusionNFT. The branches $\boldsymbol{v}^{\pm}_\theta = \boldsymbol{v}_{\mathrm{anc}} \pm \bar{\beta}(\boldsymbol{v}_\theta - \boldsymbol{v}_{\mathrm{anc}})$ are affine in $\boldsymbol{v}_\theta$ with slope $\pm\bar{\beta}$, so differentiating (NFT) and substituting the branches gives, with $r^i_{\mathrm{rvm}} = 2r^i_{\mathrm{nft}} - 1$,

$$\nabla_{\boldsymbol{v}_\theta}\mathcal{L}_{\mathrm{nft}} = r^i_{\mathrm{nft}}\,(\boldsymbol{v}^+_\theta - \boldsymbol{v}^i_{\mathrm{tgt}}) - (1 - r^i_{\mathrm{nft}})\,(\boldsymbol{v}^-_\theta - \boldsymbol{v}^i_{\mathrm{tgt}}) = \bar{\beta}\,(\boldsymbol{v}_\theta - \boldsymbol{v}_{\mathrm{anc}}) - r^i_{\mathrm{rvm}}\,(\boldsymbol{v}^i_{\mathrm{tgt}} - \boldsymbol{v}_{\mathrm{anc}}). \tag{4}$$

Inserting $\boldsymbol{v}^i_{\mathrm{tgt}} - \boldsymbol{v}_{\mathrm{anc}} = (\boldsymbol{v}^i_{\mathrm{tgt}} - \boldsymbol{v}_\theta) + (\boldsymbol{v}_\theta - \boldsymbol{v}_{\mathrm{anc}})$ rearranges the right-hand side into $r^i_{\mathrm{rvm}}\,(\boldsymbol{v}_\theta - \boldsymbol{v}^i_{\mathrm{tgt}}) + (\bar{\beta} - r^i_{\mathrm{rvm}})\,(\boldsymbol{v}_\theta - \boldsymbol{v}_{\mathrm{anc}})$, which is $\nabla_{\boldsymbol{v}_\theta}\mathcal{L}_{\mathrm{rvm}}$ at the EMA anchor $\boldsymbol{v}_{\mathrm{anc}}$ with reward $r^i = r^i_{\mathrm{rvm}}$ and anchor strength $\beta(r^i_{\mathrm{rvm}}) = \bar{\beta} - r^i_{\mathrm{rvm}}$. The two gradients agree at every $\boldsymbol{v}_\theta$ and, in this case, the scalar losses also differ only by θ-independent constants, recovering (2). □

A shared velocity-regression form Although the three objectives are written differently, their velocity-field gradients can all be represented by a common anchored velocity-regression surrogate at the noisy state $\boldsymbol{x}^i_t$:

$$\mathbb{E}_{\boldsymbol{x}^{1:G}_0 \sim \pi_{\mathrm{old}}}\left[\frac{c(r^i)}{2}\,\big\|\,\big(\boldsymbol{v}_\theta(\boldsymbol{x}^i_t, t) - \boldsymbol{v}_{\mathrm{anc}}(\boldsymbol{x}^i_t, t)\big) - A(r^i)\,\big(\boldsymbol{v}^i_{\mathrm{tgt}} - \boldsymbol{v}_{\mathrm{anc}}(\boldsymbol{x}^i_t, t)\big)\big\|_2^2\right], \tag{5}$$

where $c : \mathbb{R} \to \mathbb{R}$ is a reward-dependent scale and $A : \mathbb{R} \to \mathbb{R}$ a reward-dependent reach. Recall that $\boldsymbol{v}^i_{\mathrm{tgt}} := \boldsymbol{\epsilon}^i - \boldsymbol{x}^i_0$. The term $\boldsymbol{v}_\theta - \boldsymbol{v}_{\mathrm{anc}}$ measures how far the current velocity has moved from the anchor, while $\boldsymbol{v}^i_{\mathrm{tgt}} - \boldsymbol{v}_{\mathrm{anc}}$ is the update direction from the anchor toward the flow-matching target $\boldsymbol{v}^i_{\mathrm{tgt}}$; the reach $A(r^i)$ sets how far the regression target moves along this direction and the scale $c(r^i)$ its overall weight. Each method is thus fixed by its anchor $\boldsymbol{v}_{\mathrm{anc}}$ and reward-dependent pair (c, A), read off in Tab. 4. For the RVM row, this completed-square parameterization assumes $r_{\mathrm{rvm}} + \beta \neq 0$; the original two-term objective (RVM) remains the definition in degenerate cases. Notably, the reach of DiffusionNFT divides the signed reward by the guidance strength, so its regression target lands beyond $\boldsymbol{v}^i_{\mathrm{tgt}}$ whenever $2r^i_{\mathrm{nft}} - 1 > \bar{\beta}$.

Table 4: **Velocity-based objectives through the shared regression form** (5). With $\pi_{\mathrm{old}} := \pi_\theta$, $\beta = 1$, and $\boldsymbol{v}_{\mathrm{anc}} := \boldsymbol{v}_{\mathrm{ref}}$, RAM's gradient matches that of RVM. With $\boldsymbol{v}_{\mathrm{anc}} :=$ EMA velocity, $r_{\mathrm{rvm}} = 2r_{\mathrm{nft}} - 1$, and $\beta = \bar{\beta} - r_{\mathrm{rvm}}$, DiffusionNFT equals RVM up to θ-independent constants.

Method	Sampling policy	Anchor $\boldsymbol{v}_{\mathrm{anc}}$	Scale $c(r)$	Reach $A(r)$
RAM	π_θ	$\boldsymbol{v}_{\mathrm{ref}}$	$r_{\mathrm{ram}} + 1$	$\frac{r^i_{\mathrm{ram}}}{r_{\mathrm{ram}}+1}$
DiffusionNFT	π_{old}	EMA velocity	$\bar{\beta}$	$(2r_{\mathrm{nft}} - 1)/\bar{\beta}$
RVM	π_{old}	any	$\beta\big(r_{\mathrm{rvm}}/\beta + 1\big)$	$\frac{r_{\mathrm{rvm}}/\beta}{r_{\mathrm{rvm}}/\beta+1}$

B.3 Policy-gradient-based Loss Functionals

Policy-gradient methods incorporate reward feedback through likelihood ratios. Their connection to velocity regression depends on how these ratios are estimated and differentiated.

Trajectory-based likelihood ratios FlowGRPO (Liu et al., 2025b) applies the clipped GRPO objective to the Gaussian transition probabilities of an SDE rollout. Each transition ratio is clipped separately and weighted by the group-relative reward r^i_{grpo}. The resulting regression target depends on the sampled intermediate state, as derived in App. B.4.

ELBO-based likelihood ratios EPG and PEPG (Choi et al., 2026) instead estimate endpoint likelihood ratios using independently forward-noised samples. Applying (ELBO) at a sampled state gives

$$\hat{\rho}_\theta(\boldsymbol{x}_t, t) = \exp\big(w(t)\,\big[\|\boldsymbol{v}_{\text{old}}(\boldsymbol{x}_t, t) - \boldsymbol{v}_{\text{tgt}}\|_2^2 - \|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v}_{\text{tgt}}\|_2^2\big]\big)\,.$$

This estimate supplies the detached importance factor $\bar{\rho}_\theta$ in (EPG). We consider uniform weighting $w(t) = 1$ and adaptive weighting with a detached, sample-dependent normalizer. The adaptive choice depends on the regression residual as well as time; it is held fixed during differentiation. Reference regularization is approximated by $\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v}_{\text{ref}}(\boldsymbol{x}_t, t)\|_2^2$, following Zheng et al. (2026a); Xue et al. (2026a).

PEPG adjusts the reward weight EPG uses the group-relative reward $r^i_{\text{epg}} = r^i_{\text{grpo}}$ together with a detached importance ratio. PEPG additionally subtracts the detached log-ratio from this reward weight. Writing $\rho_\theta(\boldsymbol{x}_0) = \pi_\theta(\boldsymbol{x}_0)/\pi_{\text{old}}(\boldsymbol{x}_0)$ and $\bar{\rho}_\theta = \text{sg}[\rho_\theta]$, its objective is

$$\mathcal{L}_{\text{pepg}} = \mathbb{E}_{\boldsymbol{x}_0^{1:G}\sim\pi_{\text{old}}}\big[-\bar{\rho}_\theta(\boldsymbol{x}_0^i)\big(r^i_{\text{grpo}} - \log\bar{\rho}_\theta(\boldsymbol{x}_0^i)\big)\log\rho_\theta(\boldsymbol{x}_0^i) + \beta\,\text{kl}(\boldsymbol{x}_0^i)\big], \tag{PEPG}$$

where $\text{kl}(\boldsymbol{x}_0^i)$ is the per-sample reference regularization term, approximated by the velocity penalty above. Samples and reward weights are detached, so the reward term differentiates only through $\log\rho_\theta$; the reference penalty contributes its own gradient. The additional detached log-ratio reduces the reward weight for samples whose likelihood has increased relative to π_{old}.

Connection to RVM Substituting $\log\hat{\rho}_\theta$ into these objectives gives a reward-weighted velocity-regression update because the sampling-policy error is fixed. With $w(t) = 1$, the EPG surrogate is (EPG). It matches the RVM update, up to overall scaling, when the detached ratio is included in r^i_{rvm}, the anchor is $\boldsymbol{v}_{\text{anc}} = \boldsymbol{v}_{\text{ref}}$, and the anchor coefficients agree. In the on-policy case, the ratio is one, giving the AWM specialization under matched reward, timestep weighting, and stop-gradient conventions. PEPG retains the additional $-\log\bar{\rho}_\theta$ correction in its regression weight. These connections concern the implemented surrogate gradients; they do not equate the ELBO estimates with exact endpoint likelihoods.

B.4 Trajectory-based Objectives as Velocity Regression

Trajectory-based policy-gradient updates admit a velocity-regression representation whose target depends on the sampled transition. In this subsection, $\boldsymbol{x}^i$ denotes a complete SDE rollout. Fix a denoising time t and let expectations average over grouped rollouts $\boldsymbol{x}^{1:G} \sim \pi_{\text{old}}$ and a uniform group index i, with samples and rewards detached during differentiation. Omitting clipping, the per-step GRPO surrogate is

$$\mathcal{L}_{\text{traj}} = \mathbb{E}_{\boldsymbol{x}^{1:G}\sim\pi_{\text{old}}}\Big[-r^i_{\text{grpo}}\,\frac{\pi_\theta(\boldsymbol{x}^i_{t-\Delta t}\mid\boldsymbol{x}^i_t)}{\pi_{\text{old}}(\boldsymbol{x}^i_{t-\Delta t}\mid\boldsymbol{x}^i_t)}\Big], \tag{6}$$

whose gradient, by the identity $\nabla_\theta\pi_\theta = \pi_\theta\nabla_\theta\log\pi_\theta$, coincides with that of

$$\mathbb{E}_{\boldsymbol{x}^{1:G}\sim\pi_{\text{old}}}\Big[-r^i_{\text{grpo}}\,\text{sg}\Big[\tfrac{\pi_\theta(\boldsymbol{x}^i_{t-\Delta t}|\boldsymbol{x}^i_t)}{\pi_{\text{old}}(\boldsymbol{x}^i_{t-\Delta t}|\boldsymbol{x}^i_t)}\Big]\log\pi_\theta(\boldsymbol{x}^i_{t-\Delta t}\mid\boldsymbol{x}^i_t)\Big]. \tag{7}$$

The transition log-likelihood is available in closed form. Under the SDE samplers of FlowGRPO and DanceGRPO, one denoising step is the Gaussian

$$\boldsymbol{x}_{t-\Delta t} = \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) + \sigma_t\sqrt{\Delta t}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\mu}_\theta = \boldsymbol{x}_t - \Delta t\Big[\Big(1 + \tfrac{\sigma_t^2(1-t)}{2t}\Big)\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) + \tfrac{\sigma_t^2}{2t}\boldsymbol{x}_t\Big], \tag{8}$$

where σ_t is the noise scale of the sampler; FlowGRPO uses $\sigma_t = a\sqrt{t/(1-t)}$ with stochasticity level a, and DanceGRPO a constant. Completing the square in $\boldsymbol{v}_\theta$ gives

$$-\log\pi_\theta(\boldsymbol{x}_{t-\Delta t}\mid\boldsymbol{x}_t) = w_{\text{traj}}(t)\,\big\|\boldsymbol{v}_\theta(\boldsymbol{x}_t, t) - \boldsymbol{v}_{\text{traj}}\big\|_2^2 + C,$$
$$w_{\text{traj}}(t) = \frac{\big(2t + \sigma_t^2(1-t)\big)^2\,\Delta t}{8\,\sigma_t^2\,t^2}, \tag{9}$$

with C collecting all θ-independent terms and the induced regression target

$$\boldsymbol{v}_{\mathrm{traj}} = \frac{2t}{2t + \sigma_t^2(1-t)} \cdot \frac{\boldsymbol{x}_t - \boldsymbol{x}_{t-\Delta t}}{\Delta t} \; - \; \frac{\sigma_t^2}{2t + \sigma_t^2(1-t)}\, \boldsymbol{x}_t. \tag{10}$$

Substituting (9) into (7) yields reward-weighted velocity regression toward $\boldsymbol{v}_{\mathrm{traj}}$. This target is an affine transformation of the sampled direction $(\boldsymbol{x}_t - \boldsymbol{x}_{t-\Delta t})/\Delta t$, reducing to that direction as $\sigma_t \to 0$. For positive reward weights, the update shifts the transition mean toward the sampled next state. RVM instead uses the generated sample to define $\boldsymbol{v}_{\mathrm{tgt}}$, whose negative points toward $\boldsymbol{x}_0$ in the denoising direction of decreasing time (Fig. 2).

The sampled transition adds target noise Substitution of the sampling-policy mean into (10) gives conditional mean $\boldsymbol{v}_{\mathrm{old}}(\boldsymbol{x}_t, t)$ and the centered second moment

$$\mathbb{E}_{\boldsymbol{x}_{t-\Delta t} \sim \pi_{\mathrm{old}}(\cdot|\boldsymbol{x}_t)} \left[(\boldsymbol{v}_{\mathrm{traj}} - \boldsymbol{v}_{\mathrm{old}})(\boldsymbol{v}_{\mathrm{traj}} - \boldsymbol{v}_{\mathrm{old}})^{\mathsf{T}} \,\middle|\, \boldsymbol{x}_t, t \right] = \frac{4t^2\sigma_t^2}{\left(2t + \sigma_t^2(1-t)\right)^2 \Delta t}\, \boldsymbol{I}. \tag{11}$$

Here $\boldsymbol{v}_{\mathrm{old}}$ is evaluated at $(\boldsymbol{x}_t, t)$ and the expectation is over the sampled reverse transition. The identity follows from its Gaussian noise variance $\sigma_t^2 \Delta t\, \boldsymbol{I}$ and the coefficient of $\boldsymbol{x}_{t-\Delta t}$ in (10), before conditioning on the final reward. RVM constructs $\boldsymbol{v}_{\mathrm{tgt}} = \boldsymbol{\epsilon} - \boldsymbol{x}_0 = (\boldsymbol{x}_t - \boldsymbol{x}_0)/t$ from generated samples and independent forward noise, eliminating dependence on the sampled reverse transition. Comparisons of complete reward-weighted gradient variances additionally require consistent sampling and weighting assumptions. Xue et al. (2026a) derives a target-noise ordering under a common forward-process law.

B.5 Proximal Updates with Negative Rewards

Anchoring can make the population RVM objective well posed even when reward weights are negative. Fix the prompt and sampling policy π_{old}, and take a constant $\beta > 0$. Here, $\mathbb{E}_{\pi_{\mathrm{old}}}$ averages over independent samples $\boldsymbol{x}_0^{1:G} \sim \pi_{\mathrm{old}}$, a uniform group index i, uniform $t \in (0, 1)$, and independent Gaussian forward noise. Samples, reward weights, and the anchor are detached. We minimize (RVM) over square-integrable velocity fields, assuming bounded weights and finite second moments for $\boldsymbol{v}_{\mathrm{tgt}}^i$ and $\boldsymbol{v}_{\mathrm{anc}}$.

Conditioning on $(\boldsymbol{x}_t^i, t)$ gives a quadratic whose curvature is $\beta + \mathbb{E}_{\pi_{\mathrm{old}}}[r_{\mathrm{rvm}}^i \mid \boldsymbol{x}_t^i, t]$. If this quantity has a strictly positive uniform lower bound, the unique minimizer, up to equality almost everywhere, is

$$\boldsymbol{v}_\theta = \frac{\beta \boldsymbol{v}_{\mathrm{anc}} + \mathbb{E}_{\pi_{\mathrm{old}}}[r_{\mathrm{rvm}}^i \boldsymbol{v}_{\mathrm{tgt}}^i \mid \boldsymbol{x}_t^i, t]}{\beta + \mathbb{E}_{\pi_{\mathrm{old}}}[r_{\mathrm{rvm}}^i \mid \boldsymbol{x}_t^i, t]}, \tag{12}$$

where velocity arguments $(\boldsymbol{x}_t^i, t)$ are omitted. The first-order condition gives this expression; the denominator bound and conditional Jensen's inequality ensure square integrability. Thus, individual weights may be negative as long as the total conditional curvature remains positive.

When the anchor is the detached pre-update field, the objective has the form of a proximal subproblem with parameter β^{-1}. If the unanchored reward-weighted objective is convex in the velocity field, this is a standard convex proximal step (Parikh & Boyd, 2014). Otherwise, it defines a well-posed generalized proximal step under the strictly positive uniform lower bound on total conditional curvature stated above. A frozen reference anchor instead regularizes toward that reference. This result concerns one population update and does not establish convergence of repeated updates or neural-network optimization.

B.6 Unanchored Matching with Exponential Reward Weights

Unanchored RVM fits a KL-regularized reward optimum when its weights are exponential in the raw reward. For bounded $r(\boldsymbol{x}_0)$ and reward strength $\lambda > 0$, this optimum is

$$\pi_\lambda^\star = \operatorname*{argmin}_{\pi \ll \pi_{\mathrm{old}}} \{\mathrm{KL}(\pi \| \pi_{\mathrm{old}}) - \lambda \mathbb{E}_\pi[r(\boldsymbol{x}_0)]\} = \frac{\pi_{\mathrm{old}}(\boldsymbol{x}_0) e^{\lambda r(\boldsymbol{x}_0)}}{\mathbb{E}_{\pi_{\mathrm{old}}}[e^{\lambda r(\boldsymbol{x}_0)}]}. \tag{13}$$

Substitution rewrites the minimized objective as $\mathrm{KL}(\pi \| \pi_\lambda^\star)$ plus a constant, establishing uniqueness. Comparison with π_{old} also gives $\mathbb{E}_{\pi_\lambda^\star}[r(\boldsymbol{x}_0)] \geq \mathbb{E}_{\pi_{\mathrm{old}}}[r(\boldsymbol{x}_0)]$.

Setting $r^i_{\mathrm{rvm}} = e^{\lambda r(\boldsymbol{x}^i_0)}$ and $\beta = 0$ changes the sampling measure in (RVM) to give

$$\mathcal{L}_{\mathrm{rvm}} = \frac{\mathbb{E}_{\pi_{\mathrm{old}}}[e^{\lambda r(\boldsymbol{x}_0)}]}{2}\, \mathbb{E}_{\pi^\star_\lambda}[\|\boldsymbol{v}_\theta - \boldsymbol{v}^i_{\mathrm{tgt}}\|_2^2]. \tag{14}$$

Expectations use the forward-noising law of App. B.5, with endpoints drawn from the indicated distribution. With finite target second moments, the population minimizer is

$$\boldsymbol{v}^\star_\lambda(\boldsymbol{x}^i_t, t) = \mathbb{E}_{\pi^\star_\lambda}[\boldsymbol{v}^i_{\mathrm{tgt}} \mid \boldsymbol{x}^i_t, t] = \frac{\mathbb{E}_{\pi_{\mathrm{old}}}[e^{\lambda r(\boldsymbol{x}^i_0)} \boldsymbol{v}^i_{\mathrm{tgt}} \mid \boldsymbol{x}^i_t, t]}{\mathbb{E}_{\pi_{\mathrm{old}}}[e^{\lambda r(\boldsymbol{x}^i_0)} \mid \boldsymbol{x}^i_t, t]}. \tag{15}$$

If the model realizes this field and its ODE admits unique transport along the interpolation, exact integration from Gaussian noise generates $\pi^\star_\lambda$ (Lipman et al., 2023). The two objectives then share an optimal generated-sample distribution; their losses and parameter gradients need not coincide. This interpretation requires exponential sample-wise weights and does not apply directly to the group-relative weights used in our video experiments.

C Implementation Details

C.1 Experiment Setup

Default Settings The following defaults describe the video RVM runs; OCR settings are given in App. C.2. We fine-tune each base model with LoRA (Hu et al., 2022), keeping the pretrained backbone frozen. Fine-tuning is classifier-free-guidance-free. For video training, each generated clip contains 53 frames at 15 frames per second (fps), corresponding to approximately 3.5 seconds. Samples are drawn with a deterministic ODE sampler (Zhang & Chen, 2023; Zhang et al., 2023). Among various ODE sampler, we use a deterministic 16-step DPM-Solver-2 flow ODE (Lu et al., 2022) at guidance scale 1.0, and the loss is computed as reward-weighted velocity regression at a single noised state per generation, with no policy-ratio clipping. We optimize with AdamW (β_1=0.9, β_2=0.999, weight decay 10^{-4}, ϵ=10^{-8}) at learning rate 5×10^{-5} in bf16, with gradient clipping at norm 1.0 and a fixed seed. Each iteration draws a group of K=8 samples for each of N prompts. The default RVM run uses no anchor (β=0), so each update is the reward-weighted velocity regression alone; the reference-anchor comparison in Sec. 4.3 uses the frozen base fields described in App. C.1. Per-model settings are listed in Tab. 5.

Loss Normalization Following Choi et al. (2026) and Zheng et al. (2026a), we normalize the matching loss by the scale of its reconstruction error. For each noised sample $\boldsymbol{x}^i_t$, the velocity prediction gives a clean-sample estimate $\hat{\boldsymbol{x}}^{i,\theta}_0 = \boldsymbol{x}^i_t - t\boldsymbol{v}_\theta(\boldsymbol{x}^i_t, t)$. Since $\hat{\boldsymbol{x}}^{i,\theta}_0 - \boldsymbol{x}^i_0 = -t(\boldsymbol{v}_\theta(\boldsymbol{x}^i_t, t) - \boldsymbol{v}^i_{\mathrm{tgt}})$, we use the normalized quadratic

$$\ell^i_{\mathrm{norm}} = \frac{\|\hat{\boldsymbol{x}}^{i,\theta}_0 - \boldsymbol{x}^i_0\|_2^2/d}{\max\Big(\mathrm{sg}\Big[\|\hat{\boldsymbol{x}}^{i,\theta}_0 - \boldsymbol{x}^i_0\|_1/d\Big],\, 10^{-5}\Big)} = \frac{t^2\|\boldsymbol{v}_\theta(\boldsymbol{x}^i_t, t) - \boldsymbol{v}^i_{\mathrm{tgt}}\|_2^2/d}{\max\big(\mathrm{sg}\big[t\|\boldsymbol{v}_\theta(\boldsymbol{x}^i_t, t) - \boldsymbol{v}^i_{\mathrm{tgt}}\|_1/d\big],\, 10^{-5}\big)}, \tag{16}$$

where d is the number of latent elements in one sample. The numerator is the mean squared reconstruction error, and the denominator is its mean absolute error over all non-batch dimensions, including the temporal dimension for video. We recompute the denominator for each sample and timestep but detach it during backpropagation, so gradients act only on the quadratic numerator. The lower bound 10^{-5} avoids division by small errors. This gives a sample-dependent weighting rather than a function of t alone. In RVM, the normalized matching term is weighted by r^i_{rvm} and averaged over samples; the separate anchor penalty retains its mean squared velocity difference without this normalization.

Training and evaluation data Both video models are trained on publicly available prompt sets. For the default Wan2.1-T2V-1.3B runs, we use the prompt corpus released by DanceGRPO (Xue et al., 2025), consisting of 48,998 training prompts, which we use as released without further preprocessing. For SkyReels-I2V, we follow the image-to-video protocol of DanceGRPO and train on 27,000 prompts constructed from ConsisID (Yuan et al., 2025); since the model conditions on a reference first frame, we pre-generate one reference image per prompt with FLUX (Labs, 2024) at the reduced 640×400 (width $\times$ height) geometry used in this protocol and cache the resulting prompt–image

pairs before training. For evaluation, Wan2.1-T2V-1.3B generates one video per prompt from the extended rewrites of the 946-prompt VBench suite. We evaluate these videos using the official VBench protocol (Huang et al., 2024). For SkyReels-I2V, we use the official VBench-I2V suite (Huang et al., 2026b), comprising 1,118 promptreference-image pairs, with reference images resized and center-cropped to the evaluation geometry.

Evaluation metrics The cited VideoAlign scores for DanceGRPO and TaRoS in Tab. 1 are evaluated on 1.4k iStock prompts (Li et al., 2026b), whereas ours use the 946-prompt VBench suite. We retain these published values as references, not as a comparison on the same prompt set.

Anchor evaluation For Fig. 7, we select results from the same training window for all three anchor settings. The base reference without CFG uses $\beta = 10^{-3}$, which performed well in this setting; the reference with CFG at scale 5 uses $\beta = 10^{-2}$. Training and inference use the learned velocity without CFG.

C.2 Text-to-Image (OCR) Task

For the OCR text-rendering task of Tab. 3, we fine-tune Stable Diffusion 3.5-Medium (Esser et al., 2024) with LoRA at resolution 512×512, following the protocol of Choi et al. (2026).

Hyperparameter settings Training is CFG-free throughout. Samples are drawn with a deterministic ODE sampler (Zhang & Chen, 2023; Zhang et al., 2023). Among various ODE sampler, we use 10-step DPM-Solver-2 flow ODE at guidance scale 1.0, and the same guidance scale is used for the training and reference velocities. Each epoch draws 48 prompts from the OCR prompt set with a group of K=24 images per prompt, followed by a single gradient update over the accumulated batch. We optimize with AdamW at learning rate 5×10^{-5} with gradient clipping at norm 0.02, use the adaptive ELBO weighting $w(t)$ (App. B.3), and train for 500 epochs on 6×H100 GPUs ($\sim$41 hours). The per-sample reward weight is the mean-subtracted reward within each prompt group, and the anchor term uses the frozen reference velocity with strength $\beta = 10^{-4}$. The full settings are collected in Tab. 5.

Reward The training reward is an equal-weight sum of four components: an OCR accuracy reward computed by PaddleOCR on the rendered text, PickScore (Kirstain et al., 2023) (rescaled by $1/26$ inside the scorer), CLIPScore (Hessel et al., 2021), and HPSv2.1 (Wu et al., 2023).

Evaluation Evaluation follows Choi et al. (2026): images are sampled with 40 steps, and we report OCR accuracy together with the preference and image-quality metrics PickScore, ClipScore, HPSv2.1, Aesthetic score, and ImageReward in Tab. 3.

C.3 Video Reward

VideoAlign & HPSv3 Rewards Each reward component captures a distinct aspect of video quality. VideoAlign provides text alignment (TA), motion quality (MQ), and visual quality (VQ), which measure prompt faithfulness, temporal motion quality, and frame-level appearance, respectively. We replace VQ with HPSv3 to use a human-preference-based appearance score.

Dynamic-tracking (DT) Reward Let $\{\boldsymbol{f}_1, \ldots, \boldsymbol{f}_T\}$ be the generated frames, with $T = 53$. We evaluate $n = 8$ adjacent-frame pairs with starting indices $i_k = 1 + \mathrm{round}((k-1)(T-2)/(n-1))$, $k = 1, \ldots, n$. This uses 16 distinct frames with a within-pair stride of one frame ($1/15$ second). RAFT operates at each model's generation resolution without spatial resizing. For each pair, we estimate $\boldsymbol{u}_k = \mathrm{RAFT}(\boldsymbol{f}_{i_k}, \boldsymbol{f}_{i_k+1}) \in \mathbb{R}^{2\times H\times W}$ and summarize it by the mean magnitude of its fastest pixels,

$$m_k = \frac{1}{|S_k|} \sum_{p \in S_k} \|\boldsymbol{u}_k(p)\|_2, \tag{17}$$

where p indexes pixels and S_k collects the $\lfloor 0.05\, HW \rfloor$ pixels of largest flow magnitude, so that m_k measures how fast the moving part of the scene travels between the two frames rather than how

Table 5: **Default hyperparameters** for the RVM (reward-weighted velocity matching) runs on each base model. All runs share the optimizer, DPM-2 ODE sampler, and CFG-free setting described above; the two video runs share the reward mixture of App. C.3, while the OCR run uses the reward of App. C.2.

Hyperparameter	Wan2.1-T2V-1.3B	SkyReels-I2V	SD3.5-M (OCR)
Task	text-to-video	image-to-video	text-to-image
Resolution ($H \times W$)	480×832	400×640	512×512
Frames / fps	53 / 15	53 / 15	—
Sampler	DPM-2 ODE	DPM-2 ODE	DPM-2 ODE
Sampling steps	16	16	10
Flow-matching shift	8.0	—	—
LoRA rank r	128	32	32
LoRA α	64	64	64
Learning rate	5×10^{-5}	5×10^{-5}	5×10^{-5}
Group size K	8	8	24
Prompts / iteration	32	16	48
Optimizer updates / epoch	2	2	1
Samples / update	128	64	1152
Training epochs	90	40	500
Anchor strength β (default)	0	0	10^{-4}
Gradient clip (norm)	1.0	1.0	0.02
Guidance scale (CFG)	1.0	1.0	1.0
GPUs	8	8	6

much of the frame moves. Each m_k is then measured against the resolution-aware motion threshold $\tau = 6\min(H, W)/256$, and averaged over the pairs:

$$R_{\mathrm{dt}}(\boldsymbol{x}_0) = \frac{1}{n}\sum_{k=1}^{n} \min\Big(\frac{m_k}{\tau},\, 1\Big) \in [0, 1]. \tag{DT}$$

Second Version of the DT Reward for SkyReels-I2V For SkyReels-I2V training, we use a second version of the dynamic-tracking reward. We use a bounded window so that increasing flow magnitude does not continually increase the reward. Using the same $n = 8$ frame pairs and RAFT inputs as DT, we compute the median-subtracted flow statistic $m_{\mathrm{fg}} = \frac{1}{n}\sum_{k=1}^{n}(m_k - \tilde{m}_k)$, where $\tilde{m}_k$ is the median flow magnitude of pair k, and apply

$$R_{\mathrm{dt2}}(\boldsymbol{x}_0) = \mathrm{clip}\Big(\frac{m_{\mathrm{fg}} - \tau_{\mathrm{lo}}}{\tau_{\mathrm{mid}} - \tau_{\mathrm{lo}}}, 0, 1\Big)\, \mathrm{clip}\Big(\frac{\tau_{\mathrm{hi}} - m_{\mathrm{fg}}}{\tau_{\mathrm{hi}} - \tau_{\mathrm{mid}}}, 0, 1\Big) \in [0, 1], \tag{DT2}$$

The reward is bounded by 1, decreases above τ_{mid}, and is zero outside $[\tau_{\mathrm{lo}}, \tau_{\mathrm{hi}}]$. We use thresholds $(\tau_{\mathrm{lo}}, \tau_{\mathrm{mid}}, \tau_{\mathrm{hi}}) = (3, 5, 8)$, in the same units as m_k. For SkyReels-I2V, R_{dt2} replaces R_{dt} in the reward mixture below at the same weight.

Unless otherwise stated, video training uses the third reward mixture. Sec. 4.3 compares these three mixtures, with reference-anchor settings specified in App. C.1.

Reward mixture The default video runs use the same public reward mixture, a weighted sum of VideoAlign (Liu et al., 2025c) text alignment (TA, weight 1.5) and motion quality (MQ, 1.0), HPSv3 (Ma et al., 2025) (general 0.1 and percentile 0.1), and our RAFT-based dynamic-tracking reward (DT, 0.7); VideoAlign visual quality (VQ) is not used. Rewards are centered within each prompt group and normalized by the standard deviation across the full batch, as specified in App. B. The further video experiments in App. E use the model-specific reward settings stated there.

D Retaining Performance with Few-Step Inference

Interpreting few-step robustness RVM's robustness to fewer inference steps may be related to its endpoint-based regression structure. For $t > 0$, the target satisfies $\boldsymbol{v}^i_{\mathrm{tgt}} = (\boldsymbol{x}^i_t - \boldsymbol{x}^i_0)/t$, so velocity

matching equivalently regresses $\boldsymbol{x}_t^i - t\boldsymbol{v}_\theta(\boldsymbol{x}_t^i, t)$ toward $\boldsymbol{x}_0^i$. Ordinary squared-error regression predicts the conditional barycenter of compatible endpoints, while RVM reshapes this regression through reward weighting. We conjecture that these endpoint-directed updates may simplify denoising trajectories, helping retain performance under coarser discretization.

Evaluation setup We evaluate whether RVM retains generation performance when inference uses substantially fewer sampling steps than training. Training samples are generated with the standard 16-step sampler; the 5- and 4-step schedules are used only at inference, without further training. We compare RVM without CFG against the CFG-guided base model under identical inference schedules.

Discretization The endpoint interpretation also motivates preserving a lower-noise evaluation before the terminal update. To see this, a final Euler step from t to 0 gives $\boldsymbol{x}_t - t\boldsymbol{v}_\theta(\boldsymbol{x}_t, t)$, corresponding to a conditional barycenter under ordinary squared-error matching. At high noise, averaging over distinct compatible endpoints can obscure sample-specific details; evaluating at lower noise can reduce this ambiguity. Guided by this intuition, we sub-sample the 16-step noise grid while retaining its final lower-noise transition:

$$\begin{aligned} 4 \text{ steps}: &\quad (0.999,\ 0.889,\ 0.727,\ 0.348,\ 0), \\ 5 \text{ steps}: &\quad (0.999,\ 0.960,\ 0.889,\ 0.727,\ 0.348,\ 0). \end{aligned} \tag{18}$$

Both schedules preserve the final transition from 0.348 to 0. This interpretation motivates our schedule choice; we do not separately test the proposed mechanism or the optimality of these timesteps.

Results Fig. 8 shows that RVM retains much of its generation quality, as measured by VBench Total, as the inference budget decreases. Its VBench Total decreases from 84.12 at 16 steps to 83.69 at 5 steps and 82.29 at 4 steps, corresponding to drops of 0.43 and 1.83 points. In comparison, the CFG-guided base model decreases from 83.10 to 73.62 and 64.39, with drops of 9.48 and 18.71 points. These results demonstrate substantially greater robustness of RVM in VBench Total to reduced inference steps under the evaluated schedules. They are consistent with the trajectory-simplification hypothesis, although they do not directly measure trajectory geometry or isolate the effect of timestep placement.

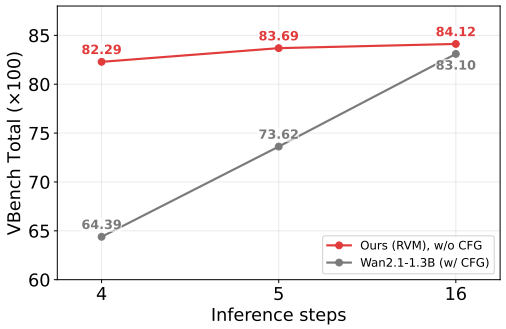


Figure 8: **Few-step inference on Wan2.1-T2V-1.3B.** VBench Total against the number of inference steps, under the same timestep schedules.

E Further Video Fine-tuning Results

We extend the video experiments to SkyReels-V1-I2V at its native spatial high resolution.

E.1 Experimental Setup

SkyReels at native spatial resolution The main-text experiments follow DanceGRPO (Xue et al., 2025) at 400×640, where the guided base scores lower than at SkyReels-V1's native 544×960 resolution (SkyReels-AI, 2025). We return to the native spatial resolution to evaluate RVM in a larger video space. Relative to App. C.3, we set the MQ weight to 0.5 and whiten reward components before mixing.

E.2 Results

RVM improves SkyReels at native spatial resolution Tab. 6 shows that RVM improves Overall from the guided base's 84.12 to 87.58 and Dynamic Degree from 38.62 to 49.19, while also improving Quality and I2V scores. RVM generates without CFG at the model's native spatial resolution.

Table 6: **Evaluation results on SkyReels-I2V at native spatial resolution** We report VBench-I2V scores on the VBench-I2V dataset. `Dyn. Degree` is the VBench dynamic-degree dimension, one of the seven quality dimensions already averaged into `Quality`.

	VBench-I2V			
Method	Dyn. Degree	Quality	I2V	Overall
SkyReels-I2V-V1 (w/o CFG)	54.10	75.85	89.74	82.80
SkyReels-I2V-V1 (w/ CFG)	38.62	76.92	91.32	84.12
RVM	49.19	80.50	94.67	87.58

F Additional Text-to-Image Experimental Results

In addition to the OCR task reported in Tab. 3, we evaluate the PickScore task of Choi et al. (2026) on DrawBench, fine-tuning SD3.5-M on the mixture of PickScore, CLIPScore, and HPSv2.1. As shown in Tab. 7, Ours (RVM) remains competitive.

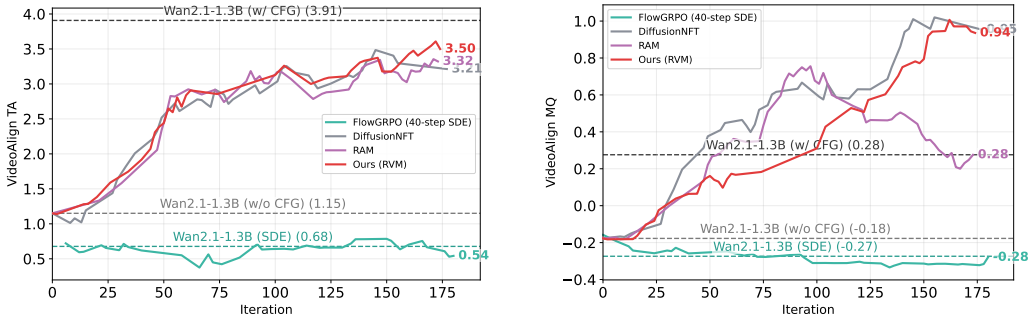
Table 7: **Evaluation Results on SD3.5-M (DrawBench)** We report the text-to-image PickScore fine-tuning task, evaluated on DrawBench with the protocol of Choi et al. (2026). Training uses PickScore, CLIPScore, and HPSv2.1. Blue shading marks the first and second highest scores per column, including ties. †cited numbers from Choi et al. (2026).

Method	PickScore	ClipScore	HPSv2.1	Aesthetic	ImgRwd
FlowGRPO†	23.50	0.280	0.316	5.90	1.29
DiffusionNFT†	23.61	0.288	0.344	6.04	1.46
PEPG†	23.68	0.296	0.325	6.06	1.45
RVM	23.30	0.289	0.333	5.82	1.38

G Additional Experimental Results

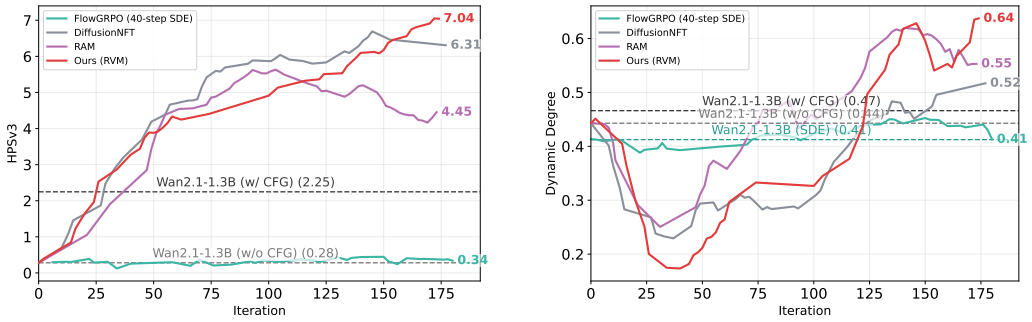
G.1 Per-reward training curves

Fig. 9 shows training curves for VideoAlign TA and MQ, HPSv3, and VBench Dynamic Degree. The velocity-matching objectives improve every component throughout training, while FlowGRPO stagnates or degrades, consistent with Tab. 1.



(a) VideoAlign text alignment (TA)

(b) VideoAlign motion quality (MQ)



(c) HPSv3

(d) VBench dynamic degree

Figure 9: **Per-reward training curves on Wan2.1-T2V-1.3B** Each panel shows one reward component across training iterations; dashed lines mark the untrained base.

G.2 Full per-dimension VBench results

Tabs. 8 and 9 expand the aggregate scores of Tabs. 1 and 2 into all 16 VBench and 10 VBench-I2V dimensions under the same protocol. Tab. 8 also includes the CFG-reference configuration from App. C.1. On Wan2.1-T2V-1.3B, the default RVM run retains the highest Quality score, while the CFG-reference run obtains the highest Overall score in this table, 84.17, with Semantic 78.88. The CFG-guided base retains the highest Semantic score. The two RVM columns use different training data and reward scaling, as detailed in App. C.1. On SkyReels-I2V, the three velocity-matching methods reach similar Overall scores, and Ours (RVM) is the only fine-tuned method that raises dynamic degree above the base.

Table 8: **Full per-dimension VBench Results on Wan2.1-T2V-1.3B** We report the per-dimension expansion of Tab. 1; all scores $\times 100$. † columns are cited numbers.

	Dimension	Wan2.1-1.3B (w/o CFG)	Wan2.1-1.3B (w/ CFG)	FlowGRPO	DanceGRPO†	TaRoS†	TaRoS-72B†	DiffusionNFT	RAM	RVM	RVM (CFG ref.)
Quality	subject consistency	91.78	95.02	91.29	95.55	95.22	95.19	97.52	96.42	98.31	97.85
	background consistency	95.14	97.72	94.77	96.83	96.77	96.85	96.06	95.25	96.22	96.79
	temporal flickering	98.95	99.54	98.95	99.42	99.41	99.35	97.93	98.62	97.25	98.26
	motion smoothness	98.48	98.27	98.19	98.15	98.28	98.23	98.25	98.43	98.44	98.22
	dynamic degree	54.17	65.28	55.56	58.33	57.67	58.33	63.89	61.11	75.00	70.83
	aesthetic quality	52.43	63.77	52.74	58.99	60.88	62.64	65.47	63.13	69.09	67.92
	imaging quality	54.75	62.97	56.29	66.61	67.92	68.49	69.92	66.99	72.02	69.16
Semantic	human action	79.00	96.00	78.00	73.00	74.00	76.00	95.00	91.00	95.00	93.00
	appearance style	21.05	21.40	20.78	20.70	20.82	21.26	19.10	19.74	19.14	19.94
	temporal style	22.47	24.62	22.35	23.51	23.29	23.15	23.28	23.24	22.97	23.71
	overall consistency	23.61	26.91	23.10	23.68	23.67	23.67	25.44	25.36	25.46	26.23
	object class	80.38	93.91	79.51	77.68	76.42	74.21	96.68	96.68	92.80	96.28
	multiple objects	51.45	82.55	48.25	57.54	59.73	63.49	88.49	81.55	82.16	86.51
	color	72.44	91.40	67.70	90.05	88.26	89.15	72.96	78.16	75.94	83.11
	spatial relationship	55.19	78.60	58.74	62.89	67.16	72.00	75.18	73.49	73.77	75.96
	scene	43.46	54.94	44.48	25.79	24.62	25.94	54.72	52.25	55.01	56.03
Quality (7)		78.59	83.72	78.67	82.81	83.23	83.66	84.37	83.35	86.09	85.49
Semantic (9)		65.73	80.64	64.87	66.06	66.26	68.15	77.28	76.33	76.28	78.88
Overall (16)		76.02	83.10	75.91	79.46	79.84	80.56	82.95	81.95	84.13	84.17

Table 9: **Full per-dimension VBench-I2V Results on SkyReels-I2V** We report the per-dimension expansion of Tab. 2; all scores $\times 100$. SkyReels-I2V-V1 is the untrained base model; [§]the same base sampled with true CFG (scale 6) and the official SkyReels-V1 negative prompt.

	Dimension	SkyReels-I2V-V1	V1 (CFG= 6)[§]	FlowGRPO	DiffusionNFT	RAM	RVM
I2V	i2v subject	91.50	92.00	88.53	96.97	97.03	95.96
	i2v background	93.48	94.56	93.18	97.15	97.65	96.59
	camera motion	31.32	33.68	32.63	30.93	29.88	30.93
Quality	subject consistency	86.17	87.85	79.63	94.96	95.29	93.27
	background consistency	91.56	93.83	91.33	96.27	96.44	95.32
	aesthetic quality	53.32	56.55	48.61	58.42	58.97	57.77
	imaging quality	64.13	63.24	58.31	68.26	68.62	68.00
	temporal flickering	96.22	96.87	96.58	97.44	97.28	96.24
	motion smoothness	97.85	97.80	97.59	98.53	98.47	97.94
	dynamic degree	64.63	47.97	41.46	47.56	51.22	72.36
I2V		87.79	88.87	86.00	93.18	93.49	92.26
Quality		75.57	75.36	69.93	79.15	79.72	80.28
Overall		81.68	82.12	77.97	86.16	86.61	86.27

G.3 Additional qualitative comparisons

Fig. 10 extends Fig. 3 to the full set of methods on Wan2.1-T2V-1.3B. Figs. 11 and 12 provide four additional Wan2.1 comparisons with the base model, and Figs. 13 and 14 give six further SkyReels-I2V examples. All videos use the protocol of Tabs. 1 and 2, and the prompt is shown once below the bottom row.

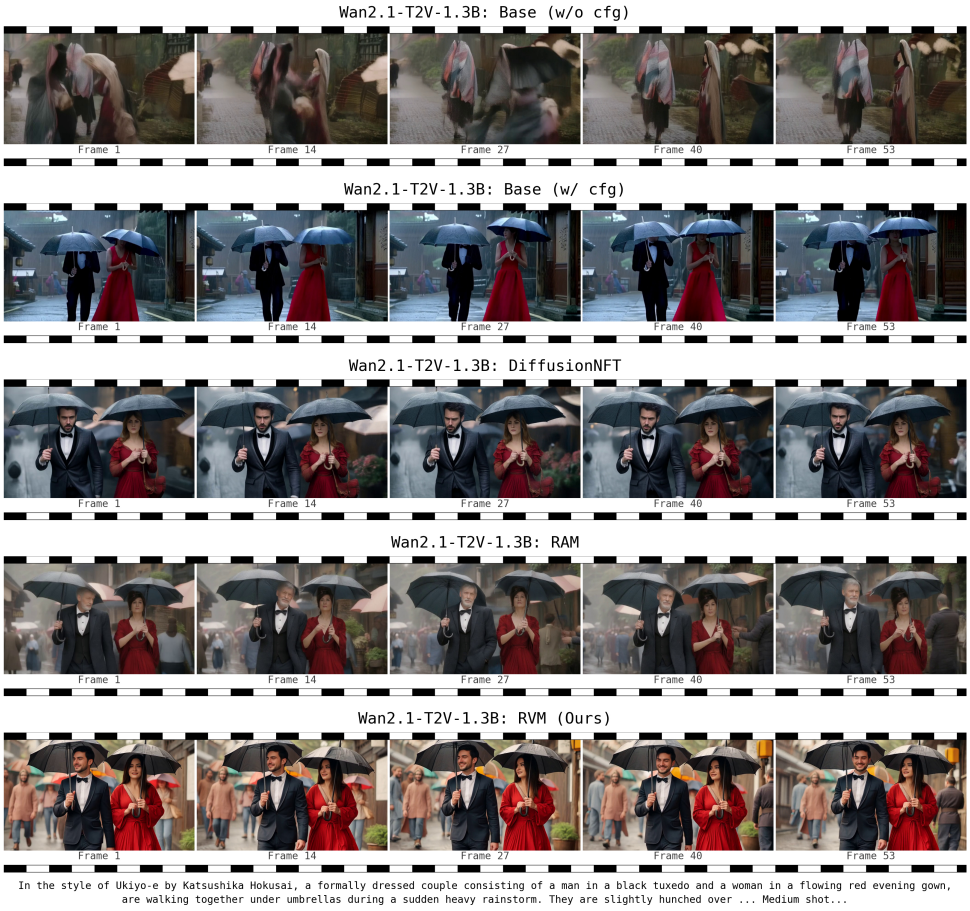


Figure 10: **Additional qualitative comparison on Wan2.1-T2V-1.3B** Frames $1, 14, 27, 40, 53$ of each method for the same prompt; top to bottom: Wan2.1-1.3B (w/o CFG), Wan2.1-1.3B (w/ CFG), DiffusionNFT, RAM, Ours (RVM).



Figure 11: **Additional qualitative comparison on Wan2.1-T2V-1.3B**. Frames $1, 14, 27, 40, 53$ of each method for two prompts; within each group of three rows, top to bottom: Wan2.1-1.3B (w/o CFG), Wan2.1-1.3B (w/ CFG), Ours (RVM).



Figure 12: **Additional qualitative comparison on Wan2.1-T2V-1.3B** Frames $1, 14, 27, 40, 53$ of each method for two prompts; within each group of three rows, top to bottom: Wan2.1-1.3B (w/o CFG), Wan2.1-1.3B (w/ CFG), Ours (RVM).

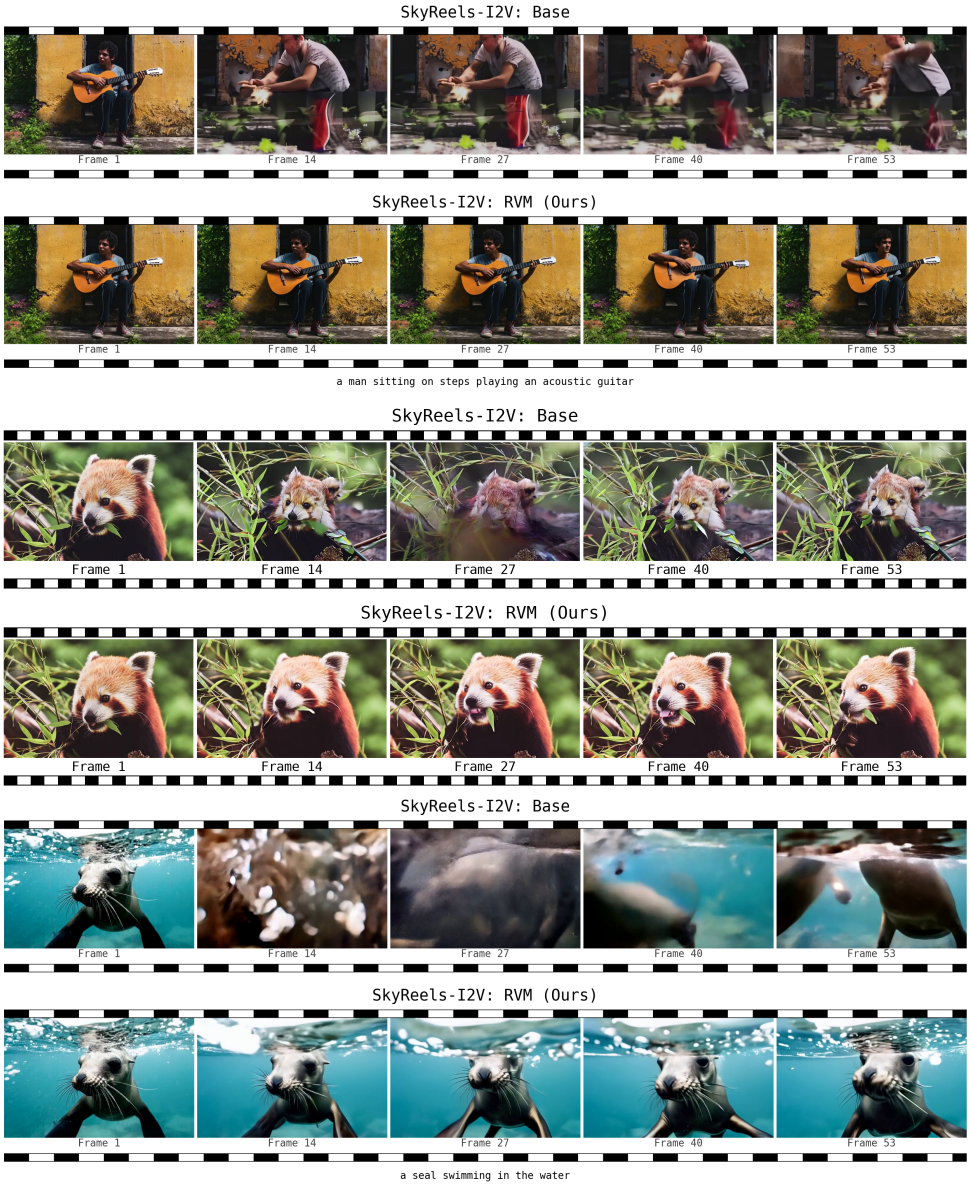


Figure 13: **Additional qualitative comparison on SkyReels-I2V** Frames $1, 14, 27, 40, 53$ for three prompts; within each pair, SkyReels-I2V-V1 (top) and Ours (RVM) (bottom).

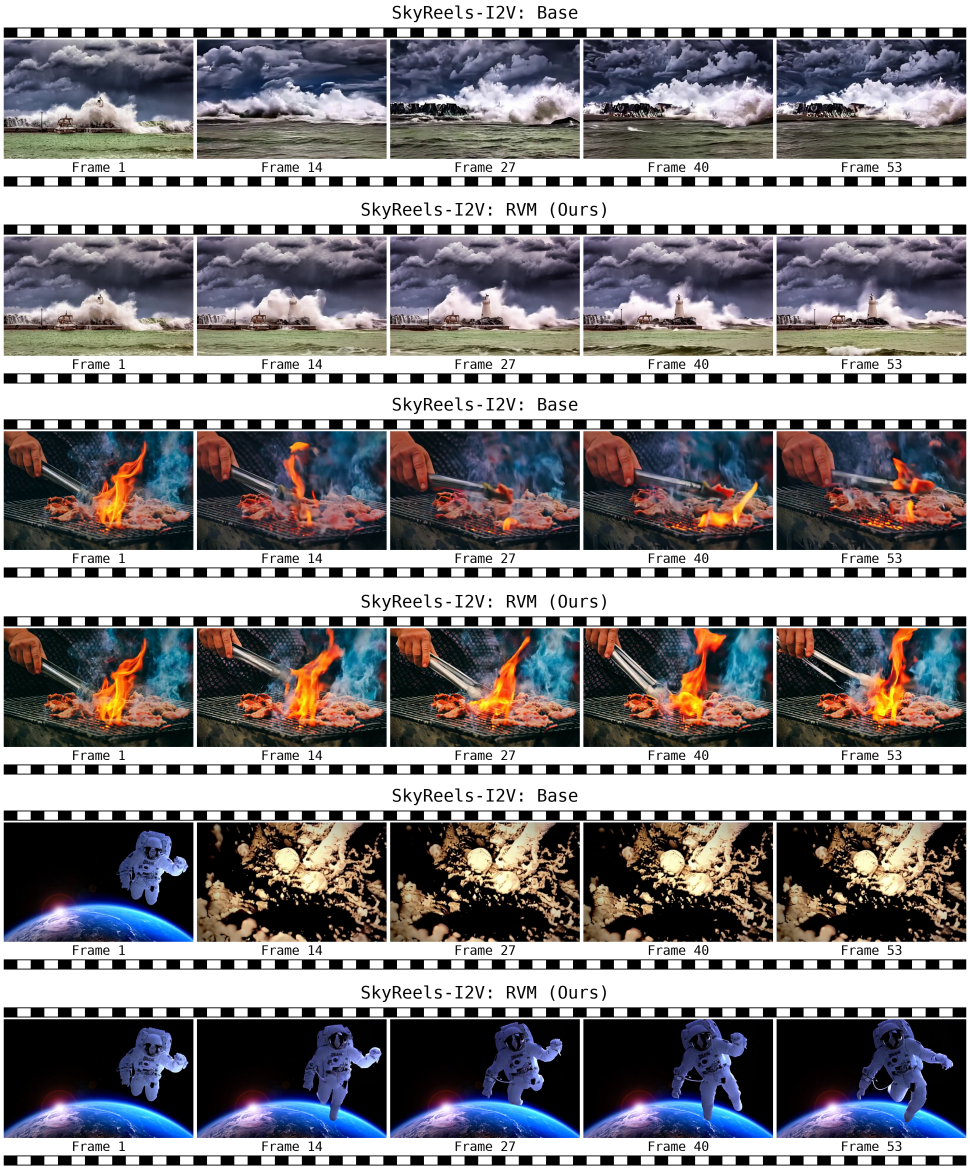


Figure 14: **Additional qualitative comparison on SkyReels-I2V** Frames $1, 14, 27, 40, 53$ for three prompts; within each pair, SkyReels-I2V-V1 (top) and Ours (RVM) (bottom).