EDPM WORKSHOP · OCTOBER 2026

Efficient Reinforcement Learning for Diffusion Models

From trajectory-based RL to native diffusion-model fine-tuning

Yongxin Chen

Georgia Institute of Technology

RL Post-training for LLMs

Maximize reward R, while staying close to the pretrained model π_{ref}

$$\max_{\theta} \; \mathbb{E}_{c\sim\mathcal{D},\, x\sim\pi_\theta(\cdot|c)}\left[R(x,c)\right] - \beta\, \mathbb{E}_{c\sim\mathcal{D}}\, \mathrm{KL}(\pi_\theta(\cdot \mid c) \,\|\, \pi_{\text{ref}}(\cdot \mid c))$$

For autoregressive LLMs the model factorizes

$$\pi_\theta(y \mid c) = \prod_{k=1}^{L} \pi_\theta(y_k \mid y_{<k}, c)$$

MDP FOR LLMS

State $s_k = (c, y_{<k})$

Action $a_k = y_k$

Transition $s_{k+1} = (s_k, a_k)$

Policy $\pi_\theta(a_k \mid s_k)$

Policy Optimization for LLMs

ROLLOUT

$$y_{1:G} \sim \pi_{\text{old}}(\cdot \mid c)$$

NORMALIZED ADVANTAGE

$$A_i = \frac{R(y_i, c) - \text{mean}_j\, R(y_j, c)}{\text{std}_j\, R(y_j, c)}$$

TOKENWISE RATIO

$$\rho_{i,k} = \frac{\pi_\theta(y_{i,k} \mid y_{i,<k}, c)}{\pi_{\text{old}}(y_{i,k} \mid y_{i,<k}, c)}$$

Group relative policy optimization (GRPO)

$$\mathcal{L}_{\text{GRPO}} = -\mathbb{E}\left[\sum_{i,k} \min(\rho_{i,k} A_i,\ \text{clip}(\rho_{i,k}, 1-\epsilon, 1+\epsilon) A_i)\right] + \beta \text{KL}(\pi_\theta \| \pi_{\text{ref}})$$

Policy gradient

$$\mathcal{L}_{\text{PG}} = -\mathbb{E}\left[\sum_{i,k} A_i\ \text{sg}(\rho_{i,k})\ \log \pi_\theta(y_{i,k} \mid y_{i,<k}, c)\right] + \beta \text{KL}(\pi_\theta \| \pi_{\text{ref}})$$

Diffusion and Flow Models in One Slide

FORWARD PROCESS

$$x_t = (1-t)x_0 + t\epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

A fixed, known interpolation between data at $t = 0$ and noise at $t = 1$.

TRAINING

$$\min_\theta \mathbb{E}_{t,x_0,\epsilon}\left[w(t)\|v_\theta(x_t,t) - (\epsilon - x_0)\|^2\right]$$

Regression onto the velocity $v = \epsilon - x_0$ that carries data to noise.

SAMPLING

$$\frac{dx_t}{dt} = v_\theta(x_t, t), \quad t : 1 \to 0$$

Integrate the learned field backward, by ODE or SDE, to draw $x_0 \sim \pi_\theta$.

x_0
clean sample

ϵ
Gaussian noise

$v_\theta(x_t, t)$
learned velocity field

$\pi_\theta(x_0)$
distribution of samples

$p_\theta(x_{t-\Delta t} \mid x_t)$
one denoising step

The pretrained model is the velocity field v_θ; the reward is evaluated only on the endpoint x_0.

GRPO for Diffusion Models

TOKEN RATIO

$$\rho_k^{\text{tok}} = \frac{\pi_\theta(y_k \mid y_{<k})}{\pi_{\text{old}}(y_k \mid y_{<k})}$$

→

DENOISING-TRANSITION RATIO

$$\rho_t^{\text{tr}} = \frac{p_\theta(x_{t-\Delta t} \mid x_t)}{p_{\text{old}}(x_{t-\Delta t} \mid x_t)}$$



Reward $R(x_0, c)$ is computed from the clean sample x_0 only

FLOWGRPO

$$\mathcal{L}_{\text{Flow-GRPO}} = -\mathbb{E}\left[\sum_{i,t} \min\left(\rho_{i,t}^{\text{tr}} A_i,\ \text{clip}(\rho_{i,t}^{\text{tr}}, 1-\epsilon, 1+\epsilon) A_i\right)\right] + \beta \text{KL}(p_\theta \| p_{\text{ref}})$$

Why This Becomes Expensive for Diffusion Models

— Intermediate states are numerical auxiliaries, and memory overhead.

MEMORY

— The objective depends on the solver and the discretization.

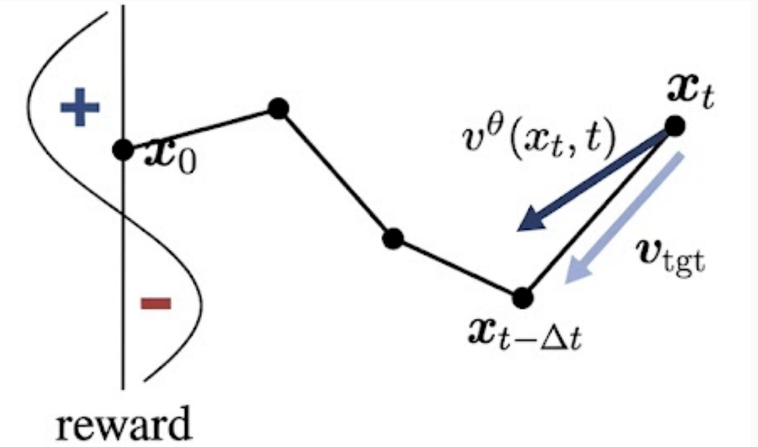
TRAINING / INFERENCE MISMATCH

— Training may require a stochastic SDE even when inference uses an ODE.

MODEL EVALUATIONS

— Transition targets point toward noisy intermediate states, not necessarily the rewarded endpoint.

LEARNING SIGNAL



PAPER 1

Rethinking the design space of reinforcement learning for diffusion models

Three Design Coordinates

Maximize reward while staying close to the reference

$$\max_{\theta} \; \mathbb{E}_{x\sim\pi_\theta}\left[R(x)\right] - \beta\,\mathrm{KL}(\pi_\theta \| \pi_{\mathrm{ref}})$$

01

Objective

GRPO, EPG, PEPG, regression

→

02

Likelihood estimator

trajectory or endpoint ELBO

→

03

Rollout sampler

stochastic SDE or deterministic ODE

Which of these three choices actually determines efficiency?

Endpoint ELBO: Variational Setup

Discrete-time schematic: let $q(x_{1:T} \mid x_0)$ discretize the fixed forward process, and let

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t)$$

be the learned reverse chain. The evidence lower bound (ELBO) is

$$\begin{aligned}
\log \pi_\theta(x_0) &\geq \mathbb{E}_q \left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)} \right] \\
&= -\mathrm{KL}(q(x_T \mid x_0) \| p(x_T)) \\
&\quad - \sum_{t=2}^{T} \mathbb{E}_q \, \mathrm{KL}(q(x_{t-1} \mid x_t, x_0) \| p_\theta(x_{t-1} \mid x_t)) \\
&\quad + \mathbb{E}_q \left[\log p_\theta(x_0 \mid x_1) \right]
\end{aligned}$$

Endpoint ELBO: Gaussian KL Becomes Regression

Under the chosen Gaussian variance and velocity parameterization,

$$\mathrm{KL}(q(x_{t-1} \mid x_t, x_0) \| p_\theta(x_{t-1} \mid x_t)) = c_t \left\| v_\theta(x_t, t) - (\epsilon - x_0) \right\|^2 + C_t,$$

where $c_t > 0$ and C_t is independent of θ.

$$\sum_t c_t \|v_\theta - v\|^2 \quad \longrightarrow \quad -\log \pi_\theta(x_0) \approx \mathbb{E}_{t,\epsilon}\left[w_{\mathrm{KL}}(t) \|v_\theta(x_t, t) - (\epsilon - x_0)\|^2 \right]$$

Path-KL weighting

$w_{\mathrm{KL}}(t)$ preserves the model-dependent ELBO interpretation.

Practical weighting

$w(t) = 1$ or adaptive $w(t)$ gives a regression surrogate; it need not remain a likelihood lower bound.

Endpoint likelihood estimation reduces to native diffusion-model training.

Likelihood Estimation Couples — or Decouples — the Rollout

	Trajectory estimator	Forward-process ELBO
Probability object	$p_\theta(x_{t-\Delta t} \mid x_t)$	surrogate marginal $\widehat{\pi}_\theta(x_0)$
Required rollout	stochastic SDE	arbitrary ODE or SDE
Stored object	full path $\{x_t\}_{t=0}^{T}$	endpoint x_0 only
Sampler in objective?	yes	no

trajectory ratio
↓ the SDE path is part of the objective

endpoint ELBO
↓ the rollout is an independent choice

Objective, likelihood estimator, and sampler are logically separate; trajectory likelihood artificially couples the latter two.

Objective Design: EPG, PEPG

EXACT POLICY GRADIENT (EPG)

$$\mathcal{L}_{\text{EPG}} = -\mathbb{E}_{x_0 \sim \pi_{\theta_{\text{old}}}} \left[A(x_0) \, \frac{\text{sg}(\pi_\theta)}{\pi_{\theta_{\text{old}}}}(x_0) \, \log \pi_\theta(x_0) \right] + \beta \text{KL}(\pi_\theta \| \pi_{\text{ref}})$$

PROXIMAL EXACT POLICY GRADIENT (PEPG)

$$\min_\theta \quad -\mathbb{E}_{x_0 \sim \pi_\theta} \left[R(x_0) \right] + \beta \, \text{KL}(\pi_\theta \| \pi_{\text{ref}}) + \frac{1}{\eta} \text{KL}(\pi_\theta \| \pi_{\text{old}})$$

$$\mathcal{L}_{\text{PEPG}} = -\mathbb{E}_{x_0 \sim \pi_{\theta_{\text{old}}}} \left[\left(A(x_0) - \log \frac{\text{sg}(\pi_\theta)}{\pi_{\theta_{\text{old}}}}(x_0) \right) \frac{\text{sg}(\pi_\theta)}{\pi_{\theta_{\text{old}}}}(x_0) \, \log \pi_\theta(x_0) \right] + \eta \text{KL}(\pi_\theta \| \pi_{\text{ref}})$$

POLICY OPTIMIZATION AS REGRESSION

$$\frac{\text{sg}(\pi_\theta)}{\pi_{\theta_{\text{old}}}}(x_0) \, \log \pi_\theta(x_0) \approx \frac{\exp\left(-\|v_\theta(x_t, t) - v(x, t)\|^2\right)}{\exp(-\|v_{\text{old}}(x_t, t) - v(x, t)\|^2)} \|v_\theta(x_t, t) - v(x, t)\|^2$$

$$\text{KL}(\pi_\theta \| \pi_{\text{ref}}) = \mathbb{E}\left[w(t) \|v_\theta(x_t, t) - v_{\text{ref}}(x, t)\|^2 \right]$$

Generate, Forget the Path, Forward-Noise

01

Fast black-box ODE or SDE rollout

→

02

Keep x_0 and $R(x_0)$

→

03

Sample (t, ϵ)

→

04

Estimate ratio and update

Memory

No trajectory storage

Sampling

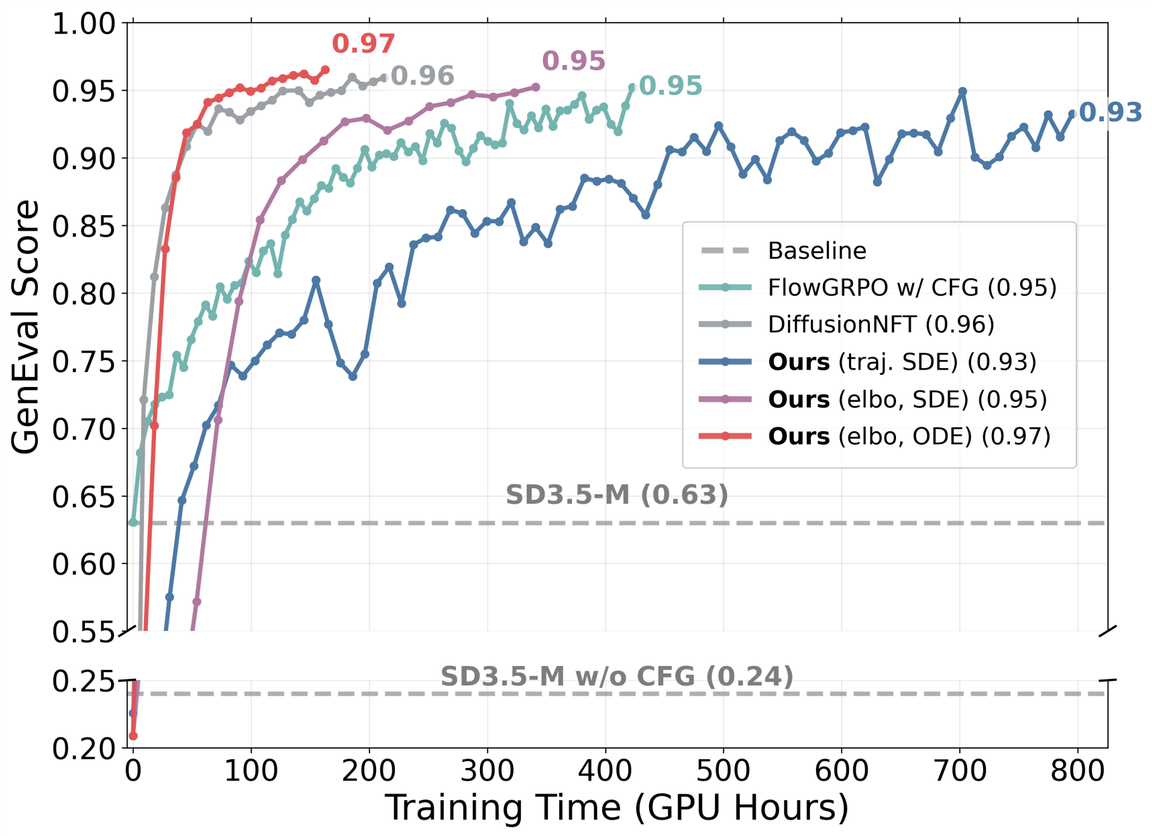
Few-step deterministic ODE

Training

Native velocity loss

Rollout, likelihood estimation, and policy optimization are now independent choices.

ELBO + ODE Is Fast



GPU hours required to reach GenEval 0.95

$\sim 4.68\times$ **fewer GPU hours:** ELBO + ODE versus FlowGRPO.

$\sim 1.24\times$ **fewer GPU hours:** ELBO + SDE versus FlowGRPO.

EPG, PEPG, PAR, and GRPO remain close once the estimator and sampler are fixed.

In these experiments, likelihood estimation and rollout matter substantially more than fine details of the policy objective.

Performance Comparison

Eval. Dataset	Model	GenEval	OCR	PickScore	ClipScore	HPSv2.1	Aesthetic	ImgRwd
Baselines	SD-XL‡	0.55	0.14	22.42	0.287	0.280	5.60	0.76
	SD3.5-L‡	0.71	0.68	22.91	0.289	0.288	5.50	0.96
	FLUX.1-Dev	0.66	0.59	22.84	0.295	0.274	5.71	0.96
	SD3.5-M	0.24	0.12	20.51	0.237	0.204	5.13	-0.58
	+ CFG	0.63	0.59	22.34	0.285	0.279	5.36	0.85
GenEval	FlowGRPO†	0.95	-	22.51	0.293	0.274	5.32	1.06
	AWM	0.89	-	22.00	0.302	0.242	4.94	0.84
	DiffusionNFT	0.95	-	**22.88**	0.303	**0.289**	5.25	1.21
	Ours	**0.96**	-	22.85	**0.305**	**0.289**	**5.33**	**1.26**
OCR	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
	Ours	-	**0.94**	**22.93**	**0.315**	**0.302**	**5.33**	**1.34**
Drawbench	FlowGRPO†	-	-	23.50	0.280	0.316	5.90	1.29
	DiffusionNFT	-	-	23.61	0.288	**0.344**	6.04	**1.46**
	Ours	-	-	**23.68**	**0.296**	0.325	**6.06**	1.45

Image Results



PAPER 2

Scaling RL for diffusion models via velocity matching

Do We Need Likelihood for RL?

AUTOREGRESSIVE MODELS

Native object: probability

$$\pi_\theta(y \mid c) = \prod_k \pi_\theta(y_k \mid y_{<k}, c)$$

Likelihood ratios are directly available, so likelihood-based policy optimization is natural.

$\longleftrightarrow$

DIFFUSION AND FLOW MODELS

Native object: score or velocity field

$$\dot{x}_t = v_\theta(x_t, t), \qquad \min_\theta \mathbb{E}\|v_\theta - v_{\text{tgt}}\|^2$$

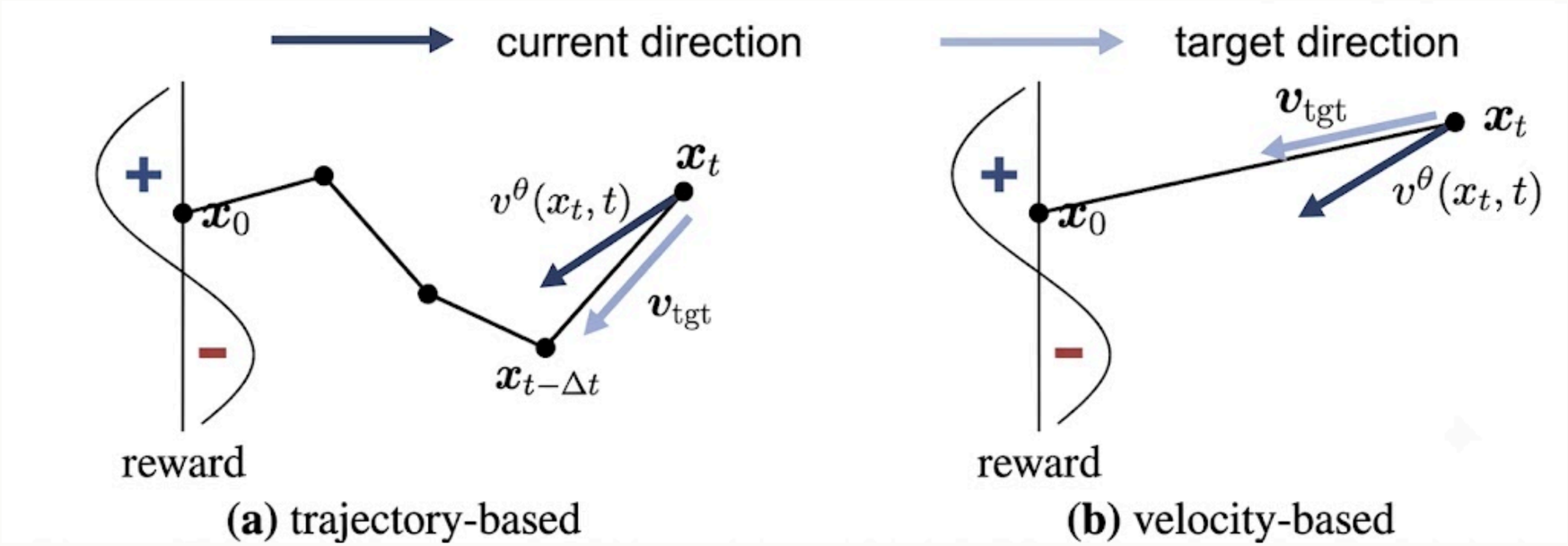
Reward feedback can act directly on the learned field instead of reconstructing a likelihood update.

Can reward feedback act directly on the model's native velocity representation?

Reward-based Velocity Matching (RVM)

Every generated sample gives a velocity target: learn from it when the reward is high, suppress it when the reward is poor, and stay anchored to a trusted reference.

$$\mathcal{L}_{\text{RVM}} = \mathbb{E}\big[\, r(x_0)\, \|v_\theta - v\|^2 + \beta(r(x_0))\, \|v_\theta - v_{\text{anc}}\|^2 \big]$$



RL changes which generated velocity targets the model should **imitate or suppress**.

RVM Unifies Velocity-Based Objectives

$$\nabla_{v_\theta} \mathcal{L}_{\mathrm{RVM}} = r\,(v_\theta - v_{\mathrm{tgt}}) + \beta(r)\,(v_\theta - v_{\mathrm{anc}})$$

METHOD	SAMPLING	ANCHOR	RVM CHOICE	RELATION
RAM	On-policy	Frozen reference	$r = r_{\mathrm{RAM}},\ \beta = 1$	Update-equivalent
DiffusionNFT	Old policy	EMA velocity	$r = 2r_{\mathrm{NFT}} - 1$ $\beta(r) = \bar{\beta} - r$	Equal up to a θ-independent constant
AWM	On-policy	Frozen reference	$r = r_{\mathrm{AWM}},\ \beta = \mathrm{const.}$	Update-equivalent*

* Under matched reward, timestep weighting, and stop-gradient conventions.

REWARD WEIGHTING R

Selects which endpoint targets to imitate—and which to suppress.

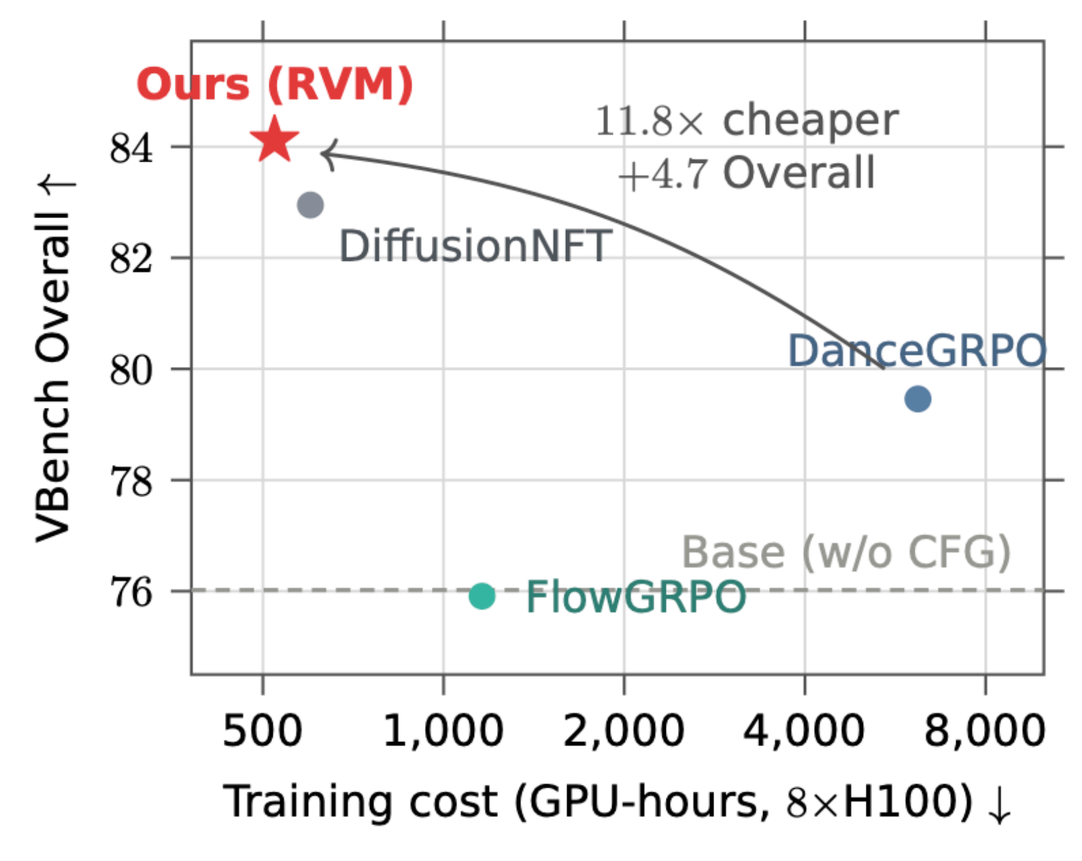
ANCHOR FIELD V_{ANC}

Controls drift around a reference, EMA, or unanchored field.

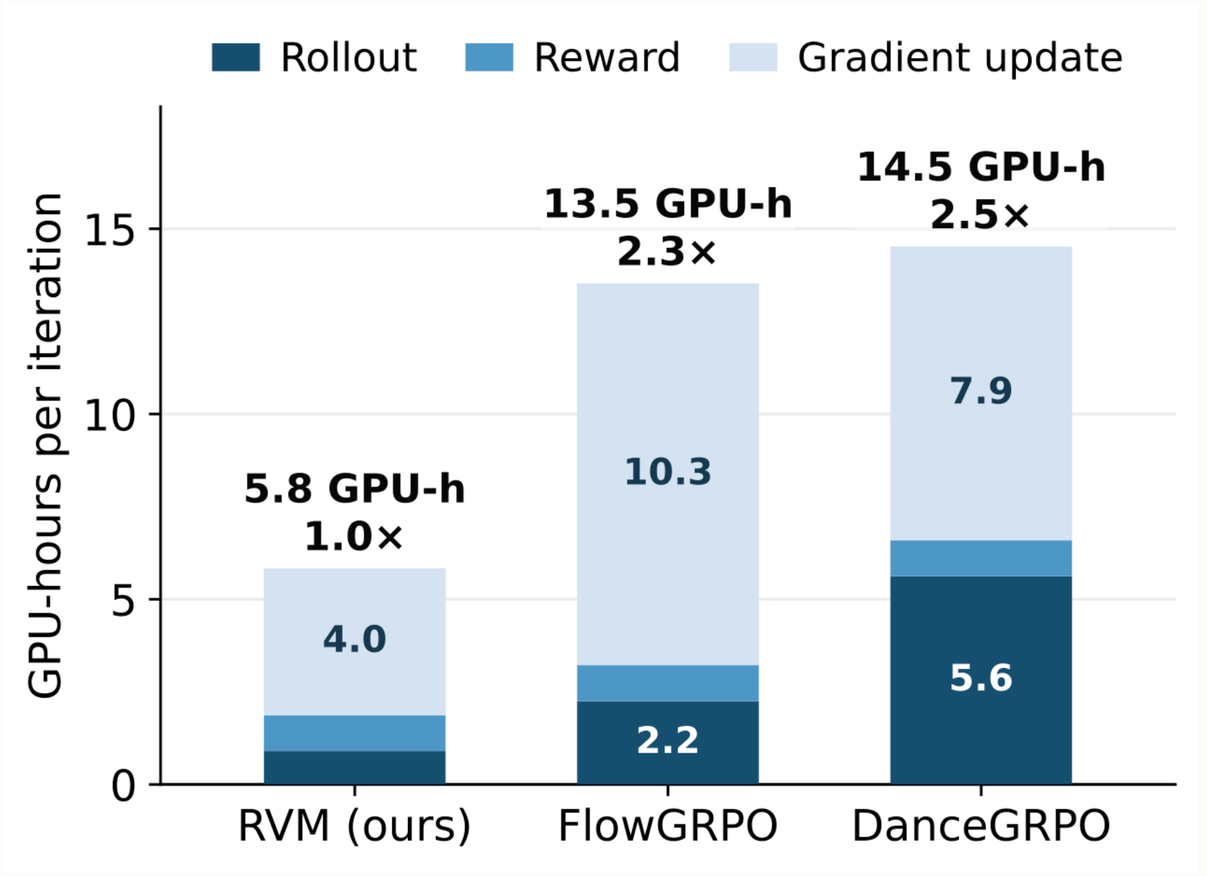
Different derivations, same basic operation: reward-shaped regression toward a velocity target, regularized around an anchor field.

Quality–Cost and Per-Iteration Efficiency

REPORTED TRAINING RUNS



PER ITERATION · BATCH 256



RVM achieves **+4.7 VBench Overall** at **11.8× lower total cost**, with each iteration **2.3–2.5× cheaper**.

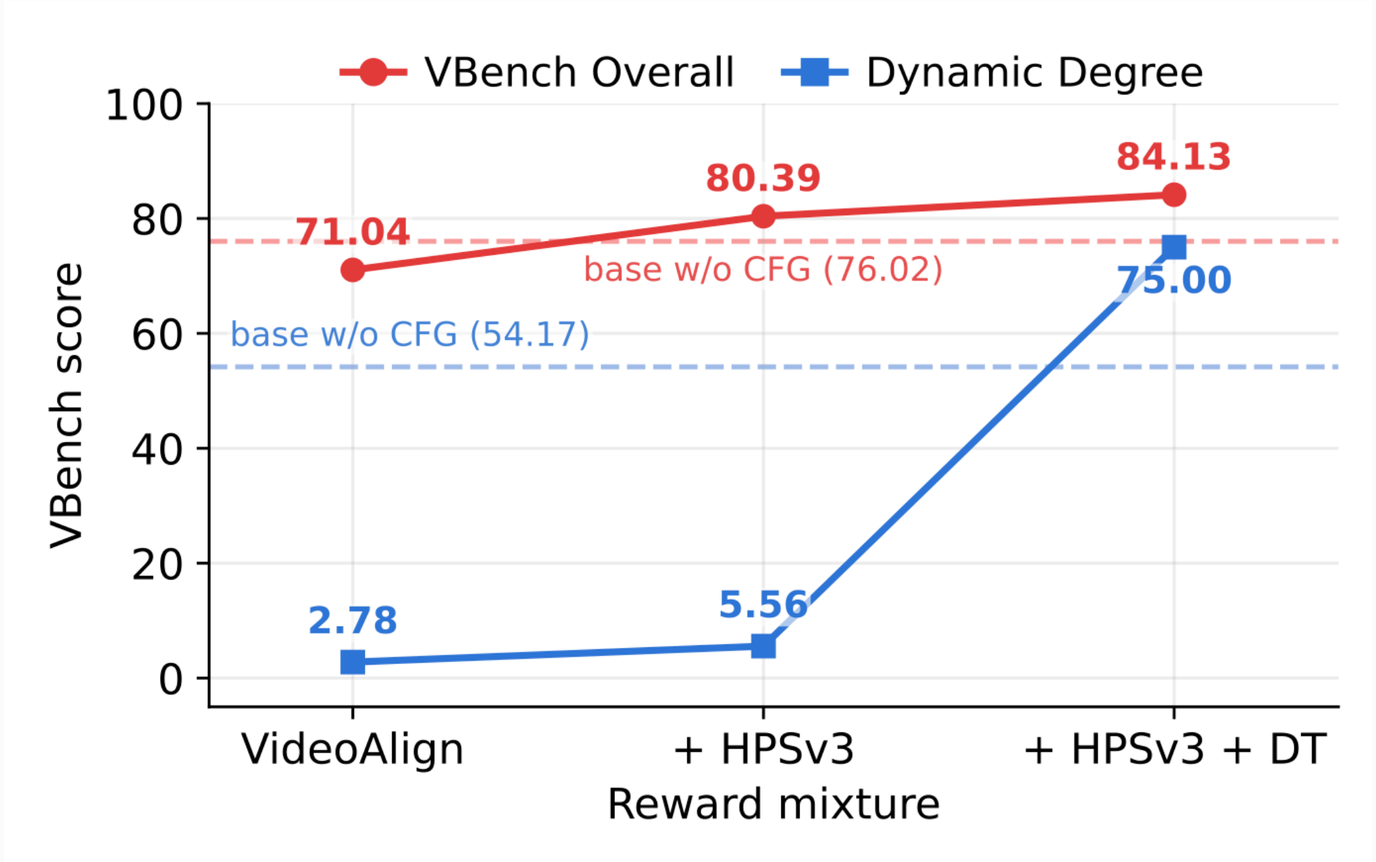
Wan2.1-T2V: Quality, Semantics, and Motion

METHOD	DYNAMIC DEGREE	QUALITY	SEMANTIC	OVERALL
Wan2.1 (w/o CFG)	54.17	78.59	65.73	76.02
Wan2.1 (w/ CFG)	65.28	83.72	**80.64**	83.10
TRAJECTORY-BASED				
FlowGRPO	55.56	78.67	64.87	75.91
DanceGRPO†	58.33	82.81	66.06	79.46
TaRoS†	57.67	83.23	66.26	79.84
TaRoS-72B†	58.33	83.66	68.15	80.56
VELOCITY-BASED				
DiffusionNFT	63.89	84.37	77.28	82.95
RAM	61.11	83.35	76.33	81.95
RVM (ours)	**75.00**	**86.09**	76.28	**84.13**

† Published values cited in the paper; evaluation protocols differ.

RVM leads in motion, quality, and overall score; CFG retains the strongest semantic score.

Once Optimization Is Efficient, Reward Design Becomes the Bottleneck

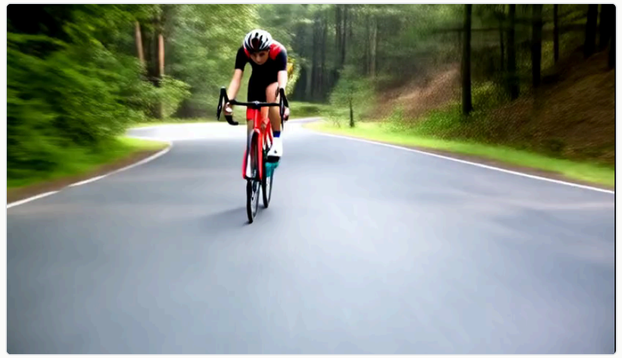
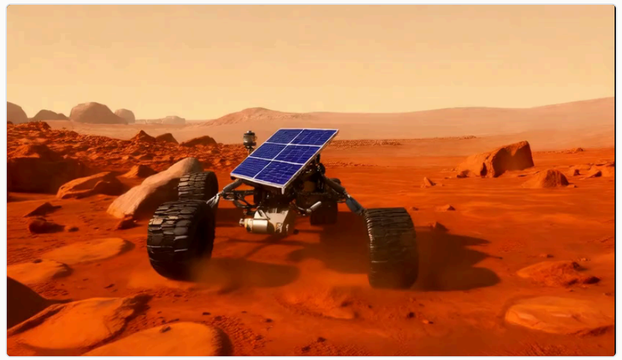


Adding explicit dynamic tracking prevents preference optimization from converging to clean but nearly static videos.

What exactly should we reward?

Video Results: Wan2.1-T2V-1.3B

TEXT-TO-VIDEO · 53 FRAMES · 16-STEP ODE · VBENCH PROMPTS

	Base (w/o CFG)	Base (CFG = 5)	RVM (ours)
“A bicycle accelerating to gain speed”			
“A Mars rover moving on Mars”			

RVM improves motion and prompt fidelity without the extra cost of classifier-free guidance.

Video Results: SkyReels-I2V

IMAGE-TO-VIDEO · VBENCH-I2V CAMERA-MOTION PROMPTS

	Base (w/o CFG)	RVM (ours)
“A large wave crashes into a lighthouse on a stormy day, camera pans left”		
“A large crowd of people walking in a shopping mall, camera pans right”		

The same objective transfers to image-to-video, where the requested camera motion is followed more faithfully.

From Trajectory Likelihood to Native Velocity Updates

LLM GRPO

$$\rho^{\text{tok}} = \frac{\pi_\theta(y_k \mid y_{<k})}{\pi_{\text{old}}(y_k \mid y_{<k})}$$

exact token ratio

→

FlowGRPO

$$\rho^{\text{tr}} = \frac{p_\theta(x_{t-\Delta t} \mid x_t)}{p_{\text{old}}(x_{t-\Delta t} \mid x_t)}$$

transition ratio

→

Endpoint ELBO RL

$$\log \hat{\rho} = \widehat{\text{ELBO}}_\theta - \widehat{\text{ELBO}}_{\text{old}}$$

marginal surrogate

→

Direct velocity matching

$$g = r(v_\theta - v) + \beta(v_\theta - v_{\text{anc}})$$

no ratio

Takeaways

01

Fine-tune velocity directly

RL for diffusion models should update the native velocity field—not reconstruct the update indirectly through likelihood estimation.

02

Reward design is essential for scaling

Once optimization is efficient, large-scale diffusion RL depends on rewards that faithfully capture the generation behavior we actually want.

Selected References

1 J. Liu et al. *Flow-GRPO: Training Flow Matching Models via Online RL*, 2025

2 J. Choi et al. *Rethinking the Design Space of Reinforcement Learning for Diffusion Models: On the Importance of Likelihood Estimation Beyond Loss Design*, 2026

3 S. Xue et al. *Advantage Weighted Matching: Aligning RL with Pretraining in Diffusion Models*, 2026

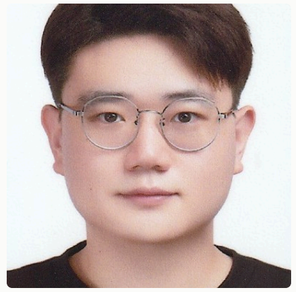
4 B. Tang et al. *V-GRPO: Online Reinforcement Learning for Denoising Generative Models Is Easier than You Think*, 2026

5 K. Zheng et al. *DiffusionNFT*, 2025

6 A. Bergmeister et al. *Reinforce Adjoint Matching*, 2026

7 J. Choi et al. *Scaling Reinforcement Learning for Diffusion Models via Velocity Matching*, 2026

ACKNOWLEDGMENT



Jaemoo Choi

Georgia Tech



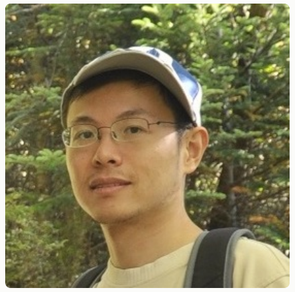
Wei Guo

Georgia Tech



Yuchen Zhu

Georgia Tech



Molei Tao

Georgia Tech



Julius Berner

NVIDIA



Arash Vahdat

NVIDIA

National Science Foundation

Thank you for your attention