STREAM4D: 4D-CONSISTENCY FOR STREAMING AUTOREGRESSIVE DIFFUSION VIDEO MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Streaming autoregressive diffusion models enable real-time, long-horizon video generation, but their training objectives optimize local frame prediction rather than the geometry and dynamics of a coherent world: long rollouts accumulate geometric drift and degrade into static or unnatural motion. Recent bidirectional approaches address this problem using reward signals built upon 3D Gaussian-Splatting reconstruction. However, a single rigid 3D reconstruction cannot model a dynamic scene, so this critic *penalizes* genuine object motion as reconstruction error and is maximized by freezing the video. This shortcut is especially detrimental in the AR setting, where each chunk can propagate an already-static configuration. In this work, we propose Stream4D, which replaces the static critic with a feed-forward 4D reconstruction reward that explicitly models scene dynamics, allowing coherent motion to receive high consistency rewards. To further guide motion magnitude and quality, we add a motion prior that rewards natural scene-flow magnitude while penalizing jitter and non-rigid artifacts. Our final recipe combines these two terms with a lightweight perceptual anchor. On the 5-second Self-Forcing and Causal-Forcing backbones and the 10-second LongLive backbone, Stream4D improves 4D reconstruction PSNR by 3.46, 5.53, and 6.76 dB, respectively, while better preserving motion dynamics and earning higher human preference. Videos can be found in the supplementary material.

1 INTRODUCTION

Autoregressive (AR) diffusion video models (Huang et al., 2025; Yang et al., 2025; Cui et al., 2025; Zhu et al., 2026; Yesiltepe et al., 2026) generate videos chunk by chunk, enabling extendable temporal horizons. Combined with recent few-step backbones, such as Wan2.1 (Wan et al., 2025), they offer a practical path toward real-time streaming video generation for embodied agents, simulators, and interactive applications. However, existing objectives do not sufficiently enforce consistent world dynamics over long rollouts. A successful rollout should maintain both **4D consistency** geometry, depth, object identity, and camera motion, and a **natural motion prior** with realistic magnitude and temporal smoothness. In practice, AR models often accumulate errors over time, causing scale drift, inconsistent depth relationships, and implausible scene or camera motion.

Recent reinforcement-learning approaches, like World-R1 (Wang et al., 2026a), attempt to improve geometric consistency using rewards derived from static 3D Gaussian Splatting (3D-GS) reconstruction. However, the rigid 3D reconstruction model prior cannot represent a dynamic scene. Any genuine object motion could appear as reconstruction error and is punished by the reward, which results in frozen scenes and permitting only camera movement, as illustrated in Figure 1. Moreover, these rewards were originally developed for bidirectional video-generation settings, whereas their limitations become more pronounced in autoregressive generation. An AR model can attend only to previously generated frames and therefore has less temporal context with which to maintain coherent scene dynamics. Under this restricted information, collapsing to a static scene becomes an even easier shortcut: once motion is suppressed in early frames, subsequent chunks can continue propagating the same rigid configuration while still receiving a high reconstruction-based reward.

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: **Static-3D rewards freeze the scene; a 4D reward keeps it moving.** Five uniformly-spaced frames from a 10.3 s LongLive rollout ("a cat running away with a fish while people chase behind"). The distilled base contains large motion but the cat and fish drift across frames; the static-3D rewards World-R1 and VideoGPA reduce the scene to a more rigid, low-motion configuration. Stream4D keeps the cat running with clear forward motion while preserving a coherent subject and scene.

To address the issues mentioned above, we introduce **Stream4D**, a reinforcement-learning framework that distills 4D-consistency and natural-motion priors into streaming AR video models. Our key insight is that recent feed-forward 4D Gaussian Splatting (4D-GS) models, such as MoVieS (Lin et al., 2026), provide a useful learned prior over how dynamic scenes evolve in space and time. Given a rollout sampled from the student policy, we reconstruct it using a pretrained 4D-GS model, re-render the resulting dynamic scene from its estimated camera trajectory, and measure perceptual agreement between the rendered frames and the original rollout. This reconstruction score rewards rollouts that can be explained by coherent geometry, motion, and camera dynamics, serving as a good 4D-consistency score.

To further improve the motion quality and guide the intensity, Stream4D adds a Gaussian motion reward peaking at the natural motion intensity that penalizes both insufficient and excessive motion. We further modulate it using temporal-smoothness and rigidity-quality factors, preventing the policy from reaching the target motion magnitude through jitter, blur, or geometrically implausible shortcuts. Finally, we include a lightweight image reward model as a perceptual anchor to preserve the visual fidelity of the original model. The three reward components are standardized independently using z-normalization and then combined additively, producing a lightweight recipe that transfers across multiple distilled AR backbones. We evaluate Stream4D on three streaming AR architectures, covering both 5s and 10s video generation. Across Self-Forcing (Huang et al., 2025), Causal-Forcing (Zhu et al., 2026), and LongLive (Yang et al., 2025) backbones, Stream4D improves 4D-PSNR from 16.88, 15.44 and 17.44 to 20.34, 20.97 and 24.20, respectively, while preserving motion and maintaining strong performance under general-purpose video reward models.

Our contributions are threefold:

- We identify accumulated 4D inconsistency as a central failure mode of distilled streaming AR video models and formulate its mitigation as a reinforcement-learning problem.
- We introduce Stream4D, a training objective that combines structured 4D reconstruction consistency, a target-centered natural-motion prior, and perceptual anchoring.
- We demonstrate consistent improvements across three distilled AR backbones, including gains of up to **6.76 dB** in 4D-PSNR, while preserving motion and general video quality over both short and long generation horizons.

2 Related Work

2.1 Autoregressive Diffusion Video Models

Diffusion-based video generation has progressed from joint-frame DiTs (Yang et al., 2024; Bao et al., 2024; Kong et al., 2024; Wan et al., 2025) toward streaming autoregressive variants that emit clips chunk-by-chunk. Early teacher-forced AR approaches (Hu et al., 2024; Gao et al., 2024) suffered from error accumulation; Diffusion Forcing (Chen et al., 2024), CausVid (Yin et al., 2025) with distribution-matching distillation (Yin et al., 2024), Self-Forcing (Huang et al., 2025), LongLive (Yang et al., 2025), Infinity-RoPE (Yesiltepe et al., 2026), and Causal-Forcing (Zhu et al., 2026) progressively closed the train–test gap via per-frame noise schedules, block-causal attention, self-rollout, and AR teachers. The resulting distilled streaming models (Sun et al., 2025) support real-time generation and double as video world models, but their supervision provides no incentive for the 3D scene to remain coherent across the rollout explicitly.

2.2 Reinforcement Learning for Image and Video Generation

Dance-GRPO (Xue et al., 2025) and Flow-GRPO (Liu et al., 2025a) perform on-policy GRPO over reverse-process trajectories, treating the denoising chain as the optimization trajectory. DiffusionNFT (Zheng et al., 2025) instead reformulates RL on the *forward* process through negative-aware fine-tuning, avoiding explicit optimization over reverse trajectories. WorldCompass (Wang et al., 2026b) extends this formulation to autoregressive world models with temporal generation dynamics, while Astrolabe (Zhang et al., 2026) further adapts it to distilled streaming AR backbones using a rolling KV cache for efficient long-horizon generation.

2.3 Geometry-Aware Rewards for Video Generation

A complementary line uses 3D reconstruction as a critic. World-R1 (Wang et al., 2026a) combines a feed-forward depth/camera estimator (Lin et al., 2025), a per-clip 3D Gaussian-Splat reconstruction, and a VLM critic (Bai et al., 2025) into a 3D-aware reward for *bidirectional* text-to-video under Flow-GRPO; VideoGPA (Du et al., 2026) uses a static 3D point-cloud reprojection score to construct preference pairs for DPO, and evaluates geometric consistency with epipolar error (Kupyn et al., 2025) and video quality with VideoReward (Liu et al., 2025b). Both reconstruct the scene as a *single rigid* 3D representation, which cannot represent a dynamic scene; consequently inter-frame object motion is structurally *penalized* as 3D inconsistency, and the reward is maximized by suppressing motion rather than merely being indifferent to it. We recast this *3D-consistency* framing as *4D-consistency* by replacing the reconstructor with MoVieS (Lin et al., 2026), a feed-forward 4D-GS model that factors the scene into a canonical point cloud plus per-frame attribute overrides and scene-flow offsets.

3 Method

Stream4D is a reinforcement-learning recipe for distilled autoregressive (AR) video models. Each rollout is graded against a feed-forward 4D Gaussian-Splatting reconstruction of itself, through 4D reconstruction consistency, a gated motion term, and a lightweight perceptual anchor. Then the rewards are summed under per-axis z-normalization and optimized with the forward-process DiffusionNFT loss.

3.1 Preliminaries

Forward-process streaming-AR RL. We build on Astrolabe (Zhang et al., 2026), which combines rolling-KV-cache generation with the DiffusionNFT (Zheng et al., 2025) update for distilled AR video models. From a shared context, we sample a group of G candidate rollouts $\{W^{(i)}\}_{i=1}^{G}$ for group-wise reward normalization. For each rollout and denoising step, let v_θ and $v_{\theta_{\text{old}}}$ denote the current and reference-model velocity predictions in the latent-video space. We write $v_\theta, v_{\theta_{\text{old}}}, v^+, v^-, v_{\text{target}} \in \mathbb{R}^d$ after flattening the latent tensor. DiffusionNFT forms positive and negative interpolated velocity predictions

$$v^+ = (1-\beta)v_{\theta_{\text{old}}} + \beta v_\theta, \qquad v^- = (1+\beta)v_{\theta_{\text{old}}} - \beta v_\theta, \tag{1}$$

and optimizes

$$\mathcal{L}_{\text{policy}} = \tilde{r} \left\| v^{+} - v_{\text{target}} \right\|_2^2 + (1 - \tilde{r}) \left\| v^{-} - v_{\text{target}} \right\|_2^2, \tag{2}$$

where v_{target} is the diffusion velocity target and $\tilde{r} \in [0, 1]$ is the normalized rollout reward. Training details are provided in Sec. 4.1.

MoVieS 4D-GS reconstruction. Given a candidate rollout $W^{(i)} = \{W_t^{(i)}\}_{t=1}^{T}$, we first estimate per-frame cameras with StreamVGGT (Zhuo et al., 2026). We then run MoVieS (Lin et al., 2026), a feed-forward 4D Gaussian-Splatting reconstructor, conditioned on the sampled frames and estimated cameras. MoVieS represents the video as a dynamic Gaussian scene: a canonical set of 3D Gaussians together with time-dependent deformation and appearance parameters that allow the scene to move and change across frames. In our pipeline, we use two outputs from this reconstruction. First, MoVieS re-renders reconstructed frames $\tilde{W}_t^{(i)}$ from the estimated camera at each time t, which we use for the reconstruction reward. Second, MoVieS provides a per-pixel 3D motion field $P \in \mathbb{R}^{T \times H \times W \times 3}$ and confidence map $\text{conf} \in [0, 1]^{T \times H \times W}$, which we use to compute motion magnitude, smoothness, and rigidity.

3.2 STREAM4D REWARD DESIGN

This section first defines the three reward components and then explains how they are combined into the final advantage.

4D-GS reconstruction R_{recon}. For each candidate rollout, we subsample 26 frames and use the MoVieS reconstruction described above to render the corresponding reconstructed frames from the estimated cameras. We compare each rendered frame with the original generated frame and define the reconstruction reward as the clipped average perceptual agreement:

$$R_{\text{recon}} = \text{clip}\left(1 - \frac{1}{T} \sum_{t} \text{LPIPS}(\tilde{W}_t^{(i)}, W_t^{(i)}), 0, 1\right). \tag{3}$$

Here $W_t^{(i)}$ is the t-th frame of candidate rollout i, $\tilde{W}_t^{(i)}$ is the corresponding frame re-rendered from the MoVieS 4D-GS reconstruction using the estimated camera, and LPIPS (Zhang et al., 2018) denotes the Learned Perceptual Image Patch Similarity distance. LPIPS measures perceptual discrepancy in a pretrained deep feature space, making it less sensitive than pixel-wise losses to small low-level misalignments while still penalizing visual reconstruction errors. Thus, the reward is high when the rollout can be explained by a coherent 4D reconstruction and low when the video contains inconsistent geometry, drifting object identity, or motion that cannot be organized into a stable dynamic scene.

Gated motion conjunction R_{mot}. Unlike the static 3D reward, the 4D reconstructor does not punish motion. However, lower-motion clips are still slightly easier to reconstruct. As shown in Fig. 2, there is a weak Spearman $\rho = -0.27$ between per-prompt 4D-PSNR and the motion-gate input m computed on the Self-Forcing base model over the held-out prompts. To address this, we add an explicit, quality-weighted motion reward that guides both motion intensity and motion quality:

$$R_{\text{mot}} = g(m) \cdot \text{smooth} \cdot \text{rigid}, \tag{4}$$

where m is the motion intensity, $g(\cdot)$ is a Gaussian gate, and smooth and rigid are quality factors that penalize jitter and spatially erratic flow. We compute these terms from the MoVieS motion field P and confidence map conf defined in Sec. 3.1. Let $v_t = P_{t+1} - P_t$ be the per-pixel 3D scene-flow velocity and $\|v_t\|$ its magnitude. The dynamic mask $\mathcal{D}$ contains the top 20% fastest

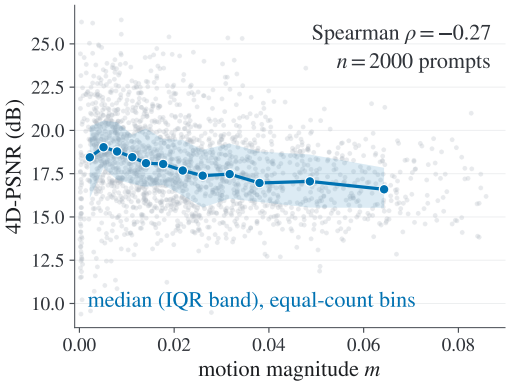


Figure 2: Unlike a static 3D reward, which punishes motion outright, our dynamic 4D-GS backbone does not penalize motion. Per-prompt 4D-PSNR only drifts weakly downward with motion magnitude m.

pixel-time entries in the clip. We write $\mathrm{mean}_{\mathcal{D}}^{\mathrm{conf}}[f] = \big(\sum_{(p,t)\in\mathcal{D}} \mathrm{conf}_{p,t} f_{p,t}\big) / \big(\sum_{(p,t)\in\mathcal{D}} \mathrm{conf}_{p,t}\big)$ for the confidence-weighted mask mean and $\mathrm{mean}_{\mathcal{D}}[f]$ for the uniform one. The clip-level motion magnitude is $m = \mathrm{mean}_{\mathcal{D}}^{\mathrm{conf}} \|v_t\|$.

The motion gate is a peaked Gaussian penalizing both static collapse and runaway blur:

$$g(m) \;=\; \exp\!\Big(-\tfrac{(m-m_{\mathrm{nat}})^2}{2\sigma^2}\Big), \tag{5}$$

where the target m_{nat} is the median per-clip m of base rollouts. We choose the Gaussian kernel because it penalizes both overly aggressive and nearly static motion, thereby encouraging motion intensity to remain within a natural range. We also evaluate alternative formulations, such as a linear motion function, in the ablation study in Sec. 4.

Smoothness penalizes excessive aggregate motion and deformation and is formulated as follows:

$$\mathrm{smooth} \;=\; \exp\!\left(-\frac{\mathrm{mean}_{\mathcal{D}}^{\mathrm{conf}}\, \|v_{t+1} - v_t\|}{m + c}\right). \tag{6}$$

where $c = 10^{-4}$ is a small constant for numerical stability.

Rigidity penalizes spatially erratic scene flow. Although real objects can deform or articulate, nearby points on the same moving object usually have locally coherent 3D velocities. We therefore compute the spatial finite differences of the per-pixel scene-flow velocity v_t within the dynamic mask $\mathcal{D}$. Large $\|\nabla_x v_t\|$ or $\|\nabla_y v_t\|$ indicates neighboring pixels moving with sharply different 3D velocities, which often corresponds to tearing, jitter, or melting artifacts. We map this roughness to a bounded quality factor:

$$\mathrm{rigid} \;=\; \exp\!\Big(-\tfrac{k_{\mathrm{rough}}}{2}\,\big(\mathrm{mean}_{\mathcal{D}}\|\nabla_x v_t\| \;+\; \mathrm{mean}_{\mathcal{D}}\|\nabla_y v_t\|\big)\Big), \quad k_{\mathrm{rough}} = 400. \tag{7}$$

This encourages locally coherent motion while still allowing smooth non-rigid or articulated dynamics.

Perceptual anchor $R_{\mathbf{hpsv2}}$. We use HPSv2 (Wu et al., 2023) to score the aesthetics of the generated frames directly. The reconstruction and motion terms constrain geometry and dynamics but leave per-frame appearance unconstrained. The perceptual anchor addresses this by tying the reward to human aesthetic preference, keeping generations close to the visual fidelity of the base model.

Z-norm reward ensembling. Following (Liu et al., 2026), our final reward sums three z-normalized axes evaluated on each candidate rollout $W^{(i)}$:

$$R(W^{(i)}) \;=\; w_{\mathrm{recon}}\, \tilde{z}\big[R_{\mathrm{recon}}\big] \;+\; w_{\mathrm{mot}}\, \tilde{z}\big[R_{\mathrm{mot}}\big] \;+\; w_{\mathrm{hpsv2}}\, \tilde{z}\big[R_{\mathrm{hpsv2}}\big], \tag{8}$$

where $\tilde{z}[\cdot]$ is the per-axis z-score over group. Following the DiffusionNFT (Zheng et al., 2025), we compute the group-centered advantage $A^{(i)} = R(W^{(i)}) - \frac{1}{G}\sum_j R(W^{(j)})$ with clipped affine normalization,

$$\tilde{r}^{(i)} \;=\; \mathrm{clip}\Big(A^{(i)}/A_{\max},\, -1,\, 1\Big)/2 + 1/2, \tag{9}$$

where $A_{\max} = 5$ is a fixed constant.

4 Experiments

4.1 Setup

Backbones and training. We evaluate Stream4D on three distilled autoregressive video backbones: Self-Forcing (Huang et al., 2025), Causal-Forcing (Zhu et al., 2026), and LongLive (Yang et al., 2025), using the Stream4D reward formulation in Eq. 8. Self-Forcing and Causal-Forcing use their native 81-frame setting, corresponding to approximately 5s at 16fps, while LongLive uses a 10.3,s window of 165 frames. A LoRA adapter is trained over the frozen base with the forward-process NFT loss of Sec. 3.1. We trained every method for 150 steps and reported the results on final checkpoints. We randomly sampled training prompts from VidProM (Wang & Yang, 2024). Full

Table 1: **Main results on the 500 motion-prominent testing set**. Metrics as defined in Sec. 4.1: MoVieS 4D reconstruction score; the 4DGT reconstruction (Xu et al., 2025) score (a reconstructor-independent cross-check, Appendix D); the order-debiased LLM judge evaluating *Motion preservation* and *Consistency*); and VideoReward (Liu et al., 2025b) paired win% vs. base.

Method	MoVieS Recon.			4DGT Recon.			LLM judge		VideoReward win%			
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	Motion↑	Consist.↑	VQ↑	MQ↑	TA↑	Ovr↑
Reference: full-diffusion T2V baselines (non-distilled, no RL)												
Wan2.1-T2V-1.3B (Wan et al., 2025)	19.15	0.828	0.258	17.14	0.614	0.399	—	—	—	—	—	—
CogVideoX-1.5-5B (Yang et al., 2024)	17.02	0.768	0.279	15.06	0.518	0.454	—	—	—	—	—	—
Distilled AR backbone: Self-Forcing (Huang et al., 2025) (5 s)												
Base (B0)	16.88	0.785	0.282	16.28	0.559	0.419	—	—	—	—	—	—
World-R1 (Wang et al., 2026a)	18.52	0.865	0.197	16.56	0.566	**0.396**	0.832	75.9	**66.0**	50.6	54.0	61.8
VideoGPA (Du et al., 2026)	17.75	0.868	0.202	16.00	0.529	0.401	0.737	60.9	51.8	39.5	**61.9**	48.0
+ **Stream4D (Ours)**	**20.34**	**0.874**	**0.195**	**17.24**	**0.650**	0.417	**0.833**	**82.2**	62.4	**62.8**	56.2	**66.2**
Distilled AR backbone: Causal-Forcing (Zhu et al., 2026) (5 s)												
Base (B0)	15.44	0.767	0.279	15.03	0.515	0.450	—	—	—	—	—	—
World-R1 (Wang et al., 2026a)	19.18	0.884	0.167	16.13	0.565	0.424	0.676	69.1	76.8	66.8	63.0	74.8
VideoGPA (Du et al., 2026)	18.04	0.878	0.195	14.86	0.503	0.495	0.706	60.4	72.0	58.3	**64.9**	66.3
+ **Stream4D (Ours)**	**20.97**	**0.893**	**0.150**	**17.21**	**0.643**	**0.421**	**0.765**	**73.9**	**77.8**	**73.2**	59.0	**76.0**
Distilled AR backbone: LongLive (Yang et al., 2025) (10.3 s)												
Base (B0)	17.44	0.844	0.231	15.45	0.521	0.428	—	—	—	—	—	—
World-R1 (Wang et al., 2026a)	22.64	**0.941**	**0.135**	17.55	0.641	0.420	0.498	54.0	81.4	67.6	66.8	78.2
VideoGPA (Du et al., 2026)	20.50	0.933	0.147	17.19	0.608	**0.374**	0.306	32.9	59.3	47.6	63.5	57.9
+ **Stream4D (Ours)**	**24.20**	0.905	0.146	**20.03**	**0.663**	0.392	**0.706**	**74.2**	**81.6**	**79.0**	**74.0**	**84.4**

hyperparameters, the reward-stack implementation, and compute details are in Appendices A and B; the full procedure is Algorithm 1.

Evaluation protocol. To stress-test the motion-preservation capabilities of our method, we construct a 500-prompt *motion-prominent* subset of VidProM using a motion filter and a keyword-based filter targeting multi-agent interactions and object motion. We also report results on 500 randomly selected prompts in Appendix G. We evaluate with four groups of metrics.

- *MoVieS Recon.* (PSNR↑ / SSIM↑ / LPIPS↓): we reconstruct each generated rollout with MoVieS 4D-GS using StreamVGGT-estimated cameras, re-render the resulting dynamic scene from the corresponding cameras, and compare each re-rendered frame with the generated input. Higher PSNR/SSIM and lower LPIPS therefore indicate that the rollout can be more faithfully explained by a coherent dynamic 4D representation.
- *4DGT Recon.* (PSNR↑ / SSIM↑ / LPIPS↓): to test whether reconstruction gains are specific to MoVieS, we repeat the reconstruction evaluation with 4DGT (Xu et al., 2025), which has a disjoint reconstruction architecture, weights, and training data. This provides a cross-reconstructor check and can prove the generability of our method.
- *LLM judge* (Motion↑, Consist. win%↑): a reward-blind, order-debiased Gemini-3.5-Flash judge compares each method against the corresponding distilled base. *Motion* is a three-way assessment of object/subject motion. We map our three tags, preserved, reduced and mostly lost, to $1/0.5/0$ and average over prompts. *Consist.* is a head-to-head win/tie/loss verdict that gives credit for improved consistency only when object motion is retained; a cleaner but frozen rollout therefore cannot win by suppressing motion. To eliminate the position bias, each pair is evaluated in both presentation orders and the two scores are averaged. Exact prompts and reproducibility details are given in Appendix E.
- *VideoReward* (Liu et al., 2025b) (VQ / MQ / TA / Overall↑): we report the per-prompt paired win-rate against the same backbone's distilled base under VideoReward's Visual Quality, Motion Quality, Text Alignment, and Overall heads.

4.2 Main Results

Table 1 reports Stream4D at ckpt-150 on the 500 motion-prominent testing set against the distilled base, World-R1 (Wang et al., 2026a) and VideoGPA (Du et al., 2026) on each of the three backbones.

Table 2: **Human study on LongLive.** We focus the human evaluation on the long-horizon LongLive backbone, where accumulated drift and the static-collapse shortcut matter most. For each pair raters pick which video shows *more natural motion* and which is *better overall* (jointly weighing motion amount, motion quality, and object consistency); win% is Stream4D's rate (ties count $\frac{1}{2}$, $>50\%$ prefers ours). For reference we list the Gemini judge's *Motion* and *Consistency* win% on the same 50 prompts.

Stream4D vs.	**Human win%↑**		**Gemini judge win%↑**	
	Motion	**Overall**	**Motion**	**Consist.**
Base	43	60	63	70
World-R1 (Wang et al., 2026a)	72	76	93	68
VideoGPA (Du et al., 2026)	87	80	95	82

4D-consistency gains transfer across all three backbones. Stream4D lifts 4D-PSNR by $+3.46$ dB on Self-Forcing , $+5.53$ dB on Causal-Forcing, and $+6.76$ dB on LongLive , with SSIM and LPIPS improving on all three. Under 4DGT (Xu et al., 2025), a reconstructor with disjoint architecture, weights, and training data from MoVieS, Stream4D again posts the best PSNR and SSIM in every backbone block, leading World-R1 by $+0.7$ / $+1.1$ / $+2.5$ dB. This indicates the 4D-reconstruction gain is not specific to MoVieS' model; it does not by itself certify metric-accurate geometry, a caveat we make precise in Appendix D. Furthermore, the vision-LLM judge gives Stream4D the best consistency score on every backbone, beating base on $82.2\,\%$ / $73.9\,\%$ / $74.2\,\%$ of prompts against World-R1's $75.9\,\%$ / $69.1\,\%$ / $54.0\,\%$, confirming the improved consistency.

Object motion is preserved; 3D method collapses it. The vision-LLM judge scores Stream4D's motion preservation at 0.83 / 0.77 / 0.71 on SF / CF / LL highest across the three backnones compared to the baselines. One can see qualitative filmstrips for all three backbones in Appendix Fig. S1, showing the failure mode directly: the static-reward baselines lock the subject in place while Stream4D keeps it moving. These results support our claim that fitting dynamic scenes with 3D representations can suppress motion, as object movement introduces ambiguous intermediate states that are harder to represent consistently, thereby motivating the use of 4D representations to better preserve scene dynamics.

Stream4D beats World-R1 on VideoReward-Overall on every backbone. Overall win-rates against base are $66.2\,\%$ / $76.0\,\%$ / $84.4\,\%$, versus World-R1's $61.8\,\%$ / $74.8\,\%$ / $78.2\,\%$. The margin is largest on the motion-quality head ($+12.2$ / $+6.4$ / $+11.4$ pp): on motion-prominent prompts, a reward that keeps motion alive wins preference over one that freezes it. These results show that Stream4D improves overall preference in addition to motion quality and consistency, providing further evidence that its gains translate to reward-model judgments aligned with human preference.

Human evaluation. To assess performance in a realistic use case, we conduct a blinded human study using the 10.3s LongLive settings. To this end, we first sample 50 prompts from our 500 motion-prominent subset and form three pairings, Stream4D vs. base, Stream4D vs. World-R1, and Stream4D vs. VideoGPA, giving 150 two-alternative forced-choice comparisons. Each trial asks which clip shows *more natural motion* and which is *better consistency* jointly weighing motion amount, motion quality, and object consistency. Then we split it into 5 parts and distribute to 5 raters. As shown in Table 2, humans prefer Stream4D over both geometry-reward baselines by a wide margin: $76\,\%$ overall vs. World-R1 and $80\,\%$ vs. VideoGPA. Full protocol and screening are in Appendix F.

4.3 Reward Ablation Study

We study the reward in three parts: dropping each of the three axes in Table 3, varying the shape of the motion term in Table 4 and sensistivity to m in Table 5

Each axis is necessary on at least one backbone; none can be dropped safely on all three. In Table 3, dropping the *motion term* (b) collapses motion on every backbone and loses the vlm consistency score to base on all three despite near-best PSNR. Dropping *reconstruction* (c) can achieve the most motion, but both reconstruction results collapse. Dropping the *perceptual anchor*

Table 3: **Reward-axis ablation:** We ablate dropping one of the three reward axes and test it on all three backbones. Each block lists the distilled base, the deployed Stream4D recipe, then the three single-axis drops. Metric conventions match Table 1.

Reward configuration	MoVieS Recon.			4DGT Recon.			LLM judge		VideoReward win%			
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	Motion↑	Consist.↑	VQ↑	MQ↑	TA↑	Ovr↑
Backbone: Self-Forcing (5 s)												
Base (no RL)	16.88	0.785	0.282	16.28	0.559	0.419	—	—	—	—	—	—
Stream4D (Ours)	20.34	0.874	0.195	17.24	0.650	0.417	0.833	82.2	62.4	62.8	56.2	66.2
− perceptual anchor (a)	22.39	0.905	0.163	18.53	0.723	0.371	0.830	81.7	66.4	64.8	52.8	66.6
− motion term (b)	24.37	0.952	0.096	19.36	0.738	0.328	0.341	31.1	63.0	56.8	44.0	59.2
− reconstruction (c)	13.98	0.692	0.342	14.17	0.466	0.476	0.983	24.2	31.6	26.4	34.6	28.2
Backbone: Causal-Forcing (5 s)												
Base (no RL)	15.44	0.767	0.279	15.03	0.515	0.450	—	—	—	—	—	—
Stream4D (Ours)	20.97	0.893	0.150	17.21	0.643	0.421	0.765	73.9	77.8	73.2	59.0	76.0
− perceptual anchor (a)	23.93	0.901	0.106	18.48	0.792	0.341	0.591	47.3	67.0	57.8	48.8	64.4
− motion term (b)†	24.19	0.862	0.094	18.89	0.776	0.329	0.453	22.4	71.6	56.2	51.0	63.4
− reconstruction (c)	14.88	0.706	0.297	14.41	0.414	0.495	0.984	45.1	45.2	50.4	50.0	49.2
Backbone: LongLive (10.3 s)												
Base (no RL)	17.44	0.844	0.231	15.45	0.521	0.428	—	—	—	—	—	—
Stream4D (Ours)	24.20	0.905	0.146	20.03	0.663	0.392	0.706	74.2	81.6	79.0	74.0	84.4
− perceptual anchor (a)	20.57	0.936	0.143	16.84	0.639	0.379	0.745	74.2	75.0	60.0	57.0	69.4
− motion term (b)	22.84	0.931	0.133	17.69	0.611	0.438	0.416	44.6	78.4	66.6	65.4	74.4
− reconstruction (c)	18.29	0.844	0.217	16.13	0.520	0.427	0.956	80.2	54.2	66.8	59.8	63.4

Table 4: **Motion-term formulation ablation:** We vary the shape of the motion term R_{mot} while keeping the reconstruction and perceptual axes fixed, on all three backbones. We test removing some components of the motion term and test different kernels for modeling motion, in addition to the Gaussian kernel. Metric conventions match Table 1.

Motion-term R_{mot}	MoVieS Recon.			4DGT Recon.			LLM judge		VideoReward win%			
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	Motion↑	Consist.↑	VQ↑	MQ↑	TA↑	Ovr↑
Backbone: Self-Forcing (5 s)												
Base (no RL)	16.88	0.785	0.282	16.28	0.559	0.419	—	—	—	—	—	—
$g(m)\cdot\text{smooth}\cdot\text{rigid}$	20.34	0.874	0.195	17.24	0.650	0.417	0.833	82.2	62.4	62.8	56.2	66.2
$\text{smooth}\cdot\text{rigid}$,	23.43	0.947	0.119	18.02	0.671	0.405	0.406	36.4	78.4	60.4	53.8	74.6
$m\cdot\text{smooth}\cdot\text{rigid}$ (e)	18.46	0.833	0.234	16.85	0.632	0.392	0.923	71.2	46.4	47.4	41.8	46.0
$g(m)$ only	23.07	0.907	0.155	18.99	0.745	0.356	0.795	75.8	64.4	65.0	52.2	64.0
$g(m)$+smooth+rigid	21.45	0.907	0.165	17.37	0.647	0.410	0.750	75.9	65.8	57.0	56.0	65.6
Backbone: Causal-Forcing (5 s)												
Base (no RL)	15.44	0.767	0.279	15.03	0.515	0.450	—	—	—	—	—	—
$g(m)\cdot\text{smooth}\cdot\text{rigid}$	20.97	0.893	0.150	17.21	0.643	0.421	0.765	73.9	77.8	73.2	59.0	76.0
$\text{smooth}\cdot\text{rigid}$,	23.14	0.922	0.116	18.37	0.741	0.338	0.396	31.2	75.0	65.8	51.0	71.0
$m\cdot\text{smooth}\cdot\text{rigid}$ (e)	22.76	0.918	0.120	17.74	0.728	0.400	0.578	61.8	69.6	68.6	49.6	68.8
$g(m)$ only	21.68	0.898	0.130	17.85	0.710	0.355	0.619	58.4	72.2	64.0	50.4	68.2
$g(m)$+smooth+rigid	21.66	0.886	0.131	18.16	0.685	0.343	0.525	44.4	74.2	61.8	51.8	68.0
Backbone: LongLive (10.3 s)												
Base (no RL)	17.44	0.844	0.231	15.45	0.521	0.428	—	—	—	—	—	—
$g(m)\cdot\text{smooth}\cdot\text{rigid}$	24.20	0.905	0.146	20.03	0.663	0.392	0.706	74.2	81.6	79.0	74.0	84.4
$\text{smooth}\cdot\text{rigid}$,	23.44	0.924	0.142	18.58	0.652	0.417	0.699	74.1	74.6	72.2	72.6	79.2
$m\cdot\text{smooth}\cdot\text{rigid}$ (e)	18.85	0.889	0.193	15.91	0.578	0.459	0.884	58.6	58.0	56.8	60.4	60.6
$g(m)$ only	20.02	0.847	0.232	18.92	0.766	0.287	0.978	64.0	37.4	64.0	50.6	50.2
$g(m)$+smooth+rigid	16.70	0.763	0.327	16.40	0.700	0.346	0.995	30.6	30.6	48.2	41.4	38.0

makes CF lose to base on vlm consistency judge and on LL it costs 3.6 dB of 4D-PSNR and 15 pp of VideoReward-Overall.

The motion term needs its peaked, multiplicative shape. In Table 4, removing the gate (d) freezes SF and CF as smooth and rigid are trivially maximized by a near-static rollout. A linear term (e) over-drives motion on SF/LL at a reconstruction and preference cost. The gate alone (f) can preserve the motion and improve the reconstruction, but it will cost the VideoReward 2.2 7.8, and 34.2 pp and is less preferred by the vlm consistency judge, indicating the low quality motion appears without these two terms. The additive split (g) is benign on SF but a net loss on CF and catastrophic on LL.

Table 5: **Sensitivity to the gate target m_{nat}** on Self-Forcing. The 0.020 row is the deployed recipe, landing on the knee of the motion-fidelity trade-off, taking the best Consistency and VideoReward-Overall.

m_{nat}	MoVieS Recon.			4DGT Recon.			LLM judge		VideoReward win% vs base			
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	Motion↑	Consist.↑	VQ↑	MQ↑	TA↑	Ovr↑
Self-Forcing base	16.88	0.785	0.282	16.28	0.559	0.419	—	—	—	—	—	—
0.010 (0.5×)	**21.10**	**0.924**	**0.135**	**17.86**	**0.672**	**0.345**	0.522	44.7	**62.8**	59.2	**65.6**	63.8
0.020 (deployed)	20.34	0.874	0.195	17.24	0.650	0.417	0.833	**82.2**	62.4	**62.8**	56.2	**66.2**
0.030 (1.5×)	18.87	0.836	0.228	17.50	0.645	0.374	**0.916**	74.4	54.0	51.0	45.6	53.2

Trade-off between motion and reconstruction. Figure 3 plots every trained variant in the reconstruction–motion plane, revealing a clear tension between preserving motion and maintaining a consistent scene. Two characteristic failure modes emerge. First, high-reconstruction/low-motion runs (upper left; e.g., rows (b) and (d)) obtain strong reconstruction scores largely by suppressing object motion. Second, high-motion/low-reconstruction runs (lower right; e.g., row (c)) preserve or even amplify motion but sacrifice scene coherence. In contrast, the deployed recipes (stars) lie on the upper-right frontier, achieving substantially stronger reconstruction without collapsing motion. This frontier behavior supports the design of our reward: the motion term prevents the trivial static optimum, while the reconstruction and perceptual terms constrain the generated dynamics to remain coherent and visually plausible.

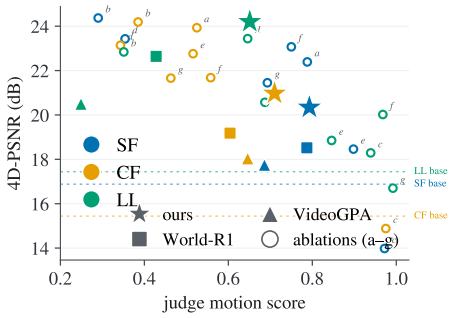


Figure 3: **Reconstruction-motion trade-off**. Deployed recipes (stars) sit on the upper-right frontier.

Sensitivity to the gate target m_{nat}. The one constant the recipe calibrates from data is the gate target $m_{\text{nat}} = 0.020$. Table 5 ablates it at 0.010 and 0.030 on Self-Forcing and shows the gate behaves as designed: motion rises monotonically from $0.52 \to 0.83 \to 0.92$ and trades off MoVieS-PSNR from $21.10 \to 20.34 \to 18.87$. Setting it as 0.020 just hits a balance between motion and consistency, gaining the best LLM judge Consist score and Video reward results.

5 Conclusion

We presented Stream4D, a reinforcement-learning recipe that moves geometric reward design for streaming AR video models from 3D- to 4D-consistency. A rigid 3D reconstruction cannot model a dynamic scene, so it penalizes motion and lets the policy earn reward by freezing the scene; a feed-forward 4D reconstruction does not penalize motion, and paired with a peaked motion gate and a light perceptual anchor it instead rewards coherent *moving* content. The single recipe transfers across three distilled backbones, improving 4D reconstruction, motion preservation, and preference win-rate over both the distilled base and a World-R1 reproduction, with a reconstructor-independent cross-check and a reward-blind judge corroborating each gain. A natural next step is a streaming 4D reconstructor to match LongLive's native horizon, and grading rollouts against explicit action or camera inputs for embodied control.

Limitations.

(i) The judge metrics come from a single vision-LLM; they are corroborated by VideoReward and the 4D metrics. (ii) The 4DGT score rules out dependence on MoVieS' training bias, but the two reconstructors still share the VGGT component.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing. All AI-assisted edits were reviewed by the authors to ensure they preserve the intended technical meaning. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the method and reward design in Section 3, with the deployed reward detailed in Appendix A, training hyperparameters and compute in Appendix B, the full training loop in Appendix C, and the evaluation protocol, LLM-judge configuration, and human-study protocol in Appendices D–F.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Fan Bao, Chendong Xiang, Gang Yue, Guande He, Hongzhou Zhu, Kaiwen Zheng, Min Zhao, Shilong Liu, Yaole Wang, and Jun Zhu. Vidu: a highly consistent, dynamic and skilled text-to-video generator with diffusion models. *arXiv preprint arXiv:2405.04233*, 2024.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. *arXiv preprint arXiv:2510.02283*, 2025.

Hongyang Du, Junjie Ye, Xiaoyan Cong, Runhao Li, Jingcheng Ni, Aman Agarwal, Zeqi Zhou, Zekun Li, Randall Balestriero, and Yue Wang. Videogpa: Distilling geometry priors for 3d-consistent video generation. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2601.23286.

Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, Jun Xiao, and Long Chen. Ca2-vdm: Efficient autoregressive video diffusion model with causal generation and cache sharing. *arXiv preprint arXiv:2411.16375*, 2024.

Haoran He, Yuxiao Ye, Jie Liu, Jiajun Liang, Zhiyong Wang, Ziyang Yuan, Xintao Wang, Hangyu Mao, Pengfei Wan, and Ling Pan. Gardo: Reinforcing diffusion models without reward hacking. *arXiv preprint arXiv:2512.24138*, 2025.

Jinyi Hu, Shengding Hu, Yuxuan Song, Yufei Huang, Mingxuan Wang, Hao Zhou, Zhiyuan Liu, Wei-Ying Ma, and Maosong Sun. Acdit: Interpolating autoregressive conditional modeling and diffusion transformer. *arXiv preprint arXiv:2412.07720*, 2024.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Orest Kupyn, Fabian Manhardt, Federico Tombari, and Christian Rupprecht. Epipolar geometry improves video generation models. *arXiv preprint arXiv:2510.21615*, 2025.

Chenguo Lin, Yuchen Lin, Panwang Pan, Yifan Yu, Tao Hu, Honglei Yan, Katerina Fragkiadaki, and Yadong Mu. MoVieS: Motion-aware 4d dynamic view synthesis in one second. In *CVPR*, 2026. arXiv:2507.10065.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

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. *arXiv preprint arXiv:2505.05470*, 2025a.

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. In *Advances in Neural Information Processing Systems*, 2025b.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, et al. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

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.

Weijie Wang, Xiaoxuan He, Youping Gu, Yifan Yang, Zeyu Zhang, Yefei He, Yanbo Ding, Xirui Hu, Donny Y Chen, Zhiyuan He, et al. World-r1: Reinforcing 3d constraints for text-to-video generation. In *International Conference on Machine Learning (ICML)*, 2026a. arXiv:2604.24764.

Wenhao Wang and Yi Yang. Vidprom: A million-scale real prompt-gallery dataset for text-to-video diffusion models. *Advances in Neural Information Processing Systems*, 37:65618–65642, 2024.

Zehan Wang, Tengfei Wang, Haiyu Zhang, Xuhui Zuo, Junta Wu, Haoyuan Wang, Wenqiang Sun, Zhenwei Wang, Chenjie Cao, Hengshuang Zhao, et al. Worldcompass: Reinforcement learning for long-horizon world models. *arXiv preprint arXiv:2602.09022*, 2026b.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023.

Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4DGT: Learning a 4D Gaussian transformer using real-world monocular videos. In *NeurIPS*, 2025. arXiv:2506.08015.

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.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024.

Hidir Yesiltepe, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. In *CVPR*, 2026. arXiv:2511.20649.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22963–22974, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Songchun Zhang, Zeyue Xue, Siming Fu, Jie Huang, Xianghao Kong, Y Ma, Haoyang Huang, Nan Duan, and Anyi Rao. Astrolabe: Steering forward-process reinforcement learning for distilled autoregressive video models. *arXiv preprint arXiv:2603.17051*, 2026.

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. *arXiv preprint arXiv:2509.16117*, 2025.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2602.02214.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. In *ICLR*, 2026. arXiv:2507.11539.

This appendix is organized as follows: Appendix A details the deployed reward; Appendix B lists training hyperparameters and compute; Appendix C gives the full training loop; Appendix D details the evaluation protocol and metrics; Appendix E documents the LLM judge (prompts, position debiasing, reproducibility); Appendix F details the human-study protocol; Appendix G reports the random-subset robustness check; and Appendix H collects qualitative comparisons. Limitations are discussed at the end of Section 5.

A Reward Implementation Details

This appendix gives the deployed-recipe details that the main paper Sec. 3.2 compresses. The reward $R(W^{(i)}) = w_{\text{recon}} \cdot \tilde{z}[R_{\text{recon}}] + w_{\text{mot}} \cdot \tilde{z}[R_{\text{mot}}] + w_{\text{hpsv2}} \cdot \tilde{z}[R_{\text{hpsv2}}]$ (Eq. 8) is computed as follows; the per-backbone axis weights $(w_{\text{recon}}, w_{\text{mot}}, w_{\text{hpsv2}})$ are $(1.0,\ 1.0,\ 0.3)$ on Self-Forcing, $(1.0,\ 0.5,\ 0.6)$ on Causal-Forcing, and $(0.8,\ 1.0,\ 0.6)$ on LongLive, and all other hyperparameters are shared.

4D-GS reconstruction backbone. StreamVGGT (Zhuo et al., 2026) predicts per-frame camera extrinsics $\{C_t\}_{t=1}^{T}$ and intrinsics from a 26-frame linearly-subsampled version of the rollout at 294×518. MoVieS (Lin et al., 2026) consumes the same subsampled frames and the predicted cameras and returns: (i) a canonical Gaussian point cloud G_{can} encoding frame-invariant geometry, (ii) per-frame Gaussian-attribute overrides (opacity, scale, color) $\{A_t\}_{t=1}^{T}$, and (iii) per-frame scene-flow offsets $\{\Delta_t\}_{t=1}^{T}$. We re-render each frame with `GaussianRenderer.render` using $G_{\text{can}} \oplus A_t \oplus \Delta_t$ at C_t, then compute frame-wise LPIPS-AlexNet against the input frame and combine as in Eq. 3. A single MoVieS forward per candidate rollout is shared between R_{recon} and R_{mot} via a per-batch cache.

Motion-quality conjunction. $R_{\text{mot}} = g(m) \cdot \text{smooth} \cdot \text{rigid}$ exactly as in Eqs. 4–7: $g(m) = \exp(-(m-m_{\text{nat}})^2/(2\sigma^2))$ with $m_{\text{nat}}=0.020$, $\sigma=0.010$ (MoVieS canonical-scene units); $\text{smooth} = \exp\big(-\text{mean}_{\mathcal{D}}^{\text{conf}} \|v_{t+1} - v_t\|/(m+\epsilon)\big)$ (Eq. 6); and $\text{rigid} = \exp\big(-\frac{k_{\text{rough}}}{2}(\text{mean}_{\mathcal{D}_x}\|\nabla_x v_t\| + \text{mean}_{\mathcal{D}_y}\|\nabla_y v_t\|)\big)$ with $k_{\text{rough}} = 400$ (Eq. 7). The scene-flow magnitude m is the within-clip confidence-weighted mean over the top-20% velocity mask; it is used only inside the reward at training time and is not one of the reported evaluation metrics.

Per-axis z-normalization. Each axis $i \in \{\text{recon}, \text{mot}, \text{hpsv2}\}$ contributes $\tilde{z}[R_i] = (R_i - \text{mean}_{\text{group}}(R_i))/\text{std}_{\text{group}}(R_i)$ to the GRPO advantage, independently per group. Per-axis z-norm decouples axis scales: e.g. HPSv2 sits at $\sim$0.20 raw while the motion conjunction sits at $\sim$0.80, but after z-norm both contribute on a common scale with weights 0.3 and 1.0.

B Training Details and Hyperparameters

Table S1 lists the full training configuration.

Training details. We fine-tune a LoRA adapter (r=α=256) over the frozen base with AdamW (η=10^{-5}), mixed-precision `bf16`, 4-step distilled timesteps, group size G=24 per prompt, rolling window L=21, frame-sink S=3, and NFT trust-region β=0.1. Group advantages over Eq. 8 are mapped to the NFT reward $\tilde{r} \in [0,1]$ exactly as in Sec. 3.2 (Eq. 9). We inherit Astrolabe's stabilization recipe: an EMA-updated old policy θ_{old} for the negative reference, a selective KL penalty (He et al., 2025), and a conditional reference-policy reset on KL drift. At training time, StreamVGGT (Zhuo et al., 2026) predicts cameras on a 26-frame linearly-subsampled version of each candidate rollout at 294×518; the LPIPS-AlexNet comparison of Eq. 3 runs between the MoVieS re-render and the input video at 480×832; HPSv2 (Wu et al., 2023) runs on the generated frames directly.

Compute. The reward stack is a single StreamVGGT+MoVieS forward per candidate rollout (26 frames at 294×518), measured at $\approx$7 s/rollout; because that one forward is shared between the reconstruction and motion axes via a per-batch cache, the motion-quality conjunction adds negligible marginal cost over a reconstruction-only reward. Each backbone trains for 150 RL steps at effective batch 384 (16 prompts $\times$ G=24 rollouts), i.e. $\approx$57,600 scored rollouts. End-to-end LoRA fine-tuning to the reported ckpt-150 costs $\approx$690 / 770 / 635 GPU-hours on Self-Forcing / Causal-Forcing / LongLive. Rollout sampling dominates per-step wall-clock. This is a deliberately lightweight recipe: a LoRA adapter over a frozen 4-step distilled base, one shared

reconstruction forward per candidate, and no per-backbone tuning beyond the three reward-axis weights.

Table S1: Stream4D training hyperparameters. A single training-hyperparameter set is used across Self-Forcing, Causal-Forcing, and LongLive; only the three reward-axis weights vary per backbone.

Module	Hyperparameter	Value
Backbone	Architecture	Causal Wan 2.1 (Self-Forcing / Causal-Forcing / LongLive)
	Video resolution ($H \times W$)	480×832
LoRA fine-tuning	Rank r / scaling α	256 / 256
	Dropout	0.0
	Gradient checkpointing	True
Optimization	Hardware	4 nodes $\times$ 8$\times$H200, DDP
	Precision	bf16
	Optimizer	AdamW (β_1=0.9, β_2=0.999)
	Learning rate / weight decay	10^{-5} / 10^{-4}
	Max gradient norm	1.0
	Distillation timesteps T	4, sampled from $\{1000, 750, 500, 250\}$
NFT-GRPO	Trust-region β	0.1
	Per-axis z-normalization	enabled
	Dynamic-phase schedule	disabled
Rollout	Group size G per prompt	24
	Rolling window L / frame-sink S	21 / 3
	Window selection	Random
Reward backbone	Camera estimator	StreamVGGT (Zhuo et al., 2026)
	4D Gaussian-Splatting reconstructor	MoVieS (Lin et al., 2026) (26-frame subsample, 294×518)
	Recon comparison	LPIPS-AlexNet between re-render and input
	Perceptual anchor	HPSv2 (Wu et al., 2023) on input rollout frames
Reward stack	Axes	$\{R_{\text{recon}}, R_{\text{mot}}, R_{\text{hpsv2}}\}$ (Eq. 8)
	Per-axis weights ($w_{\text{recon}}, w_{\text{mot}}, w_{\text{hpsv2}}$)	SF (1.0, 1.0, 0.3); CF (1.0, 0.5, 0.6); LL (0.8, 1.0, 0.6)
	Gauss-gate (m_{nat}, σ)	(0.020, 0.010)
	Rigidity constant k_{rough}	400

C Algorithm

Algorithm 1 gives the full Stream4D training loop with the deployed three-axis reward (Eq. 8). The reward backbone is a single MoVieS forward per candidate rollout, shared between the reconstruction and motion-quality axes via a per-batch reward cache.

D Evaluation Protocol Details

Evaluation sets. The main-paper results (Tables 1, 3, and 4) use a 500-prompt motion-prominent subset of VidProM (Wang & Yang, 2024), disjoint from the training prompts. We choose VidProM rather than the World-R1 evaluation set because VidProM's prompts include substantial moving-subject content (people walking, vehicles, non-rigid foliage) where our reward is designed to operate; the World-R1 set is dominated by camera-tour-through-static-scenes prompts that bias toward rigid-scene methods.

Model versions. All learned components are public releases used unmodified, with identical weights at training and evaluation time: MoVieS (Lin et al., 2026) official release checkpoint (`movies_ckpt.safetensors`); StreamVGGT (Zhuo et al., 2026) public checkpoint (HuggingFace `lch01/StreamVGGT`); 4DGT (Xu et al., 2025) released full model (`4Dgt_full.pth`, level-of-detail config `tlod-l3`); VideoReward (Liu et al., 2025b) released Qwen2-VL-7B checkpoint; HPSv2 (Wu et al., 2023) checkpoint `HPS_v2.1_compressed.pt` (v2.1). The LLM judge's model string and run dates are given in Appendix E.

4D-PSNR / SSIM / LPIPS. We re-run MoVieS 4D Gaussian-Splatting on each generated rollout at evaluation time, using StreamVGGT for camera estimation followed by MoVieS reconstruction over 26 frames at a resolution of 294×518. We then re-render the reconstructed 4D scene from the estimated camera corresponding to each input frame and compare the resulting rendering $\hat{\mathbf{I}}_t$ against the original generated frame $\mathbf{I}_t$. We report standard image-space PSNR, SSIM, and LPIPS-AlexNet for each frame and average the resulting scores over the full rollout.

Algorithm 1 Stream4D: Streaming Forward-Process RL with a 4D-Consistency Reward

Require: Distilled AR policy π_θ, behavior $\pi_{\theta_{\text{old}}}$, KL reference $\pi_{\theta_{\text{ref}}}$; camera estimator (StreamVGGT); 4D-GS reconstructor (MoVieS); prompts $\mathcal{D}$; rolling window L, frame sink S, total windows N; reward weights $(w_{\text{recon}}, w_{\text{mot}}, w_{\text{hpsv2}})$

Ensure: Optimized policy π_θ

Initialize $\theta_{\text{old}} \leftarrow \theta$, $\theta_{\text{ref}} \leftarrow \theta$, risk buffer $\mathcal{B} \leftarrow \emptyset$, $\rho \leftarrow \rho_0$
for each epoch k **do**
  Sample prompts $\{c_i\}_{i=1}^{B} \sim \mathcal{D}$
  // Phase 1: Full N-window streaming rollout, G candidates each
  for $n = 1$ to N **do**
    Form context $\mathcal{C}_n$ from S-frame sink and L rolling frames
    Decode G candidates $\{W_n^{(i,j)}\}_{j=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathcal{C}_n, c_i)$ (shared prefix)
    Update KV cache with chosen W_n (one j per rollout)
  end for
  // Phase 2: 4D-consistency reward over the rollout
  for each rollout (i, j) **do**
    $C_{1:T} \leftarrow \text{StreamVGGT}(W_{1:N}^{(i,j)})$ *// per-frame cameras, 26-frame subsample*
    $(G_{\text{can}}, A_{1:T}, \Delta_{1:T}, P, \text{conf}) \leftarrow \text{MoVieS}(W_{1:N}^{(i,j)}, C_{1:T})$ *// canonical 4D-GS + scene flow*
    Re-render $\tilde{W}_{1:T}^{(i,j)}$ from $G_{\text{can}} \oplus A_t \oplus \Delta_t$ at C_t
    $R_{\text{recon}}^{(i,j)} \leftarrow$ LPIPS score on $(\tilde{W}, W)$ *// Eq. 3*
    $R_{\text{mot}}^{(i,j)} \leftarrow g(m) \cdot \text{smooth} \cdot \text{rigid}$ from (P, conf) *// Eq. 4*
    $R^{(i,j)} \leftarrow w_{\text{recon}} \tilde{z}[R_{\text{recon}}] + w_{\text{mot}} \tilde{z}[R_{\text{mot}}] + w_{\text{hpsv2}} \tilde{z}[R_{\text{hpsv2}}]$ *// Eq. 8*
  end for
  Compute group advantages $A^{(i,j)} \leftarrow R^{(i,j)} - \frac{1}{G} \sum_h R^{(i,h)}$
  // Reward uncertainty via rank disagreement (selective KL)
  $\mathcal{M}^{(i,j)} \leftarrow$ high-uncertainty mask from rank disagreement (He et al., 2025)
  // Phase 3: Window-local NFT update with rollout-level advantage
  for each training window $n \in \{1, \ldots, N\}$ **do**
    Re-roll forward to step n; detach KV cache of $W_{<n}$ as constant
    for each mini-batch over $(W_n^{(i,j)}, A^{(i,j)}, \mathcal{M}^{(i,j)})$ **do**
      Sample $t \sim \mathcal{U}(\mathcal{T}_{\text{distill}})$, $W_n^t \leftarrow (1-t)W_n + t\epsilon$, $v_{\text{target}} \leftarrow \epsilon - W_n$
      $v_\theta, v_{\theta_{\text{old}}}, v_{\theta_{\text{ref}}} \leftarrow \pi_\cdot(W_n^t, t, \mathcal{C}_n, c)$
      $v^+ \leftarrow \beta v_\theta + (1-\beta) v_{\theta_{\text{old}}}$, $v^- \leftarrow (1+\beta) v_{\theta_{\text{old}}} - \beta v_\theta$
      $\tilde{r}^{(i,j)} \leftarrow \text{clip}(A^{(i,j)} / A_{\max}, -1, 1)/2 + 1/2$
      $\mathcal{L}_{\text{policy}} \leftarrow \tilde{r} \, \|v^+ - v_{\text{target}}\|^2 + (1-\tilde{r}) \, \|v^- - v_{\text{target}}\|^2$
      $\mathcal{L}_{\text{KL}} \leftarrow \frac{1}{|\mathcal{M}|} \sum_{(i,j):\mathcal{M}^{(i,j)}=1} \|v_\theta^{(i,j)} - v_{\theta_{\text{ref}}}^{(i,j)}\|^2$
      $\theta \leftarrow \theta - \eta \nabla_\theta (\mathcal{L}_{\text{policy}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}})$
    end for
  end for
  if $\mathcal{L}_{\text{KL}} > \tau_{\text{KL}}$ **or** $k - k_{\text{last}} > K_{\max}$ **then**
    $\theta_{\text{ref}} \leftarrow \theta$, $k_{\text{last}} \leftarrow k$
  end if
  $\theta_{\text{old}} \leftarrow \gamma \, \theta_{\text{old}} + (1-\gamma) \, \theta$
end for

For PSNR, we first compute the mean-squared error over all RGB pixels:

$$\text{MSE}_t = \frac{1}{3HW} \left\| \mathbf{I}_t - \hat{\mathbf{I}}_t \right\|_2^2. \tag{10}$$

Assuming RGB values are normalized to $[0, 1]$, the per-frame PSNR is

$$\text{PSNR}_t = 10 \log_{10} \left(\frac{1}{\text{MSE}_t} \right). \tag{11}$$

We report 4D-PSNR by averaging the per-frame PSNR values:

$$\text{4D-PSNR} = \frac{1}{T} \sum_{t=1}^{T} \text{PSNR}_t. \tag{12}$$

Thus, "4D-PSNR" does not denote a new PSNR formulation; rather, it refers to standard PSNR computed after fitting and re-rendering the rollout with the dynamic 4D Gaussian-Splat representation.

SSIM measures local structural similarity between $\mathbf{I}_t$ and $\hat{\mathbf{I}}_t$. For corresponding local image windows x and y, SSIM is computed as

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{13}$$

where μ_x and μ_y denote the local means, σ_x^2 and σ_y^2 denote the local variances, σ_{xy} denotes the local covariance, and C_1 and C_2 are numerical-stability constants. We spatially average the local SSIM values to obtain a per-frame score and then average across frames:

$$\text{4D-SSIM} = \frac{1}{T}\sum_{t=1}^{T} \mathrm{SSIM}\left(\mathbf{I}_t, \hat{\mathbf{I}}_t\right). \tag{14}$$

Higher PSNR and SSIM indicate better reconstruction fidelity.

For LPIPS, we use the AlexNet-based perceptual distance. Each image pair is passed through a pretrained AlexNet feature extractor, and normalized deep features from multiple layers are compared using the learned LPIPS channel weights. Denoting the normalized feature representation at layer l by $\tilde{\phi}_l(\cdot)$ and its learned channel-wise weighting by $\mathbf{w}_l$, the per-frame LPIPS distance can be written as

$$\mathrm{LPIPS}_t = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| \mathbf{w}_l \odot \left(\tilde{\phi}_l(\mathbf{I}_t)_{h,w} - \tilde{\phi}_l(\hat{\mathbf{I}}_t)_{h,w} \right) \right\|_2^2, \tag{15}$$

where H_l and W_l denote the spatial dimensions of the feature map at layer l. We report the average LPIPS distance across frames:

$$\text{4D-LPIPS} = \frac{1}{T}\sum_{t=1}^{T} \mathrm{LPIPS}_t. \tag{16}$$

Unlike PSNR and SSIM, lower LPIPS indicates better perceptual agreement.

This evaluation uses the same reconstruction backbone as the training reward, so it directly measures the quantity optimized by the policy: how faithfully a generated rollout can be explained by a temporally dynamic 3D representation. We refer to it as "4D" reconstruction because MoVieS uses time-varying per-frame Gaussian attributes and scene flow, yielding a representation over both space and time. On the same generated rollouts, 4D reconstruction achieves approximately 5–7 dB higher PSNR than its static-3DGS counterpart, since moving content is in-distribution for the dynamic reconstructor.

What 4DGT does and does not control for. We call this metric *reconstructor-independent*, not fully independent, and state its scope precisely. It swaps the *reconstruction* model with 4DGT, so a gain that shows up under both is not an artifact of MoVieS' particular inductive biases. Yet, they both obtained cameras from the same StreamVGGT estimator.

VideoReward. We use the Qwen2-VL-7B VideoReward (Liu et al., 2025b) model, which scores video along three heads: Visual Quality (VQ, appearance and visual fidelity), Motion Quality (MQ, motion naturalness and temporal consistency), and Text Alignment (TA, adherence to the prompt), plus an Overall score. In the main tables we report the per-prompt paired win-rate against the same backbone's distilled base: the fraction of prompts on which the method's video scores higher than the base's under each head.

Vision-LLM judge. The two LLM-judge columns in the main tables come from a reward-blind vision-LLM (Gemini-3.5-Flash). *Motion* is a three-way verdict on object/subject motion (preserved= 1 / reduced= $\frac{1}{2}$ / mostly lost= 0), averaged over prompts to a score in $[0, 1]$. *Consistency win%* is a head-to-head verdict (win= 1 / tie= $\frac{1}{2}$ / loss= 0 vs. base) requiring both kept motion and better consistency, averaged and centered at $50\,\%$; a frozen rollout is always a loss, so consistency bought by suppressing motion earns no credit. All reported numbers are position-debiased by averaging the two presentation orders. Full judge prompts, the debiasing protocol, and reproducibility checks are given in Appendix E.

Baseline reproductions. World-R1 and VideoGPA target bidirectional T2V models, so no published checkpoints exist for the distilled-AR setting; both baselines are our reproductions on the same bases, trainer, and prompt data as our method. Our World-R1 reproduction also includes its periodic dynamic-only training phase, with each cycle consisting of 20 standard training steps followed by 10 dynamic-only steps.

E LLM-JUDGE DETAILS

This appendix documents the vision-LLM judge behind the *Motion* and *Consistency win%* columns of Tables 1, 3, and 4: the exact prompts, the decoding configuration, the position-debiasing protocol, and reproducibility checks.

Setup. The judge is Gemini-3.5-Flash, called via the REST API with temperature 0 and a constrained JSON response schema with a three-way verdict enum plus a one-line free-text reason. Each call attaches two videos inline as mp4, sampled at 5 fps. The judge never sees the reward, the scene-flow metric, the method names, or which video is ours; the prompts label the two clips only as "BASE (reference)" and "EVALUATED."

Motion judge prompt. The three-way motion verdict uses the following prompt, verbatim; {prompt} is the text-to-video prompt and {base_label} / {eval_label} are filled with "Video 1" / "Video 2" according to the current presentation order. The verdict labels are neutral comparisons rather than loaded terms, so the judge is not primed toward a conclusion.

```
You are judging two AI-generated videos made from the SAME text prompt.

Text prompt:
"""{prompt}"""

Two videos are attached: Video 1 (first) and Video 2 (second).
- The BASE (reference) video is {base_label}.
- The video being EVALUATED is {eval_label}.

Compare the amount of MOTION in the EVALUATED video ({eval_label}) against the
BASE video ({base_label}). "Motion" means genuine object/subject motion (people
walking, cars driving, animals moving, crowds, explosions...) - NOT camera
pans/zooms over a frozen scene, and NOT flickering/texture noise.

Answer with exactly one verdict (about the EVALUATED video relative to the BASE):
  "motion_preserved"   = the EVALUATED video shows a similar amount of (or more)
                         object motion than the BASE
  "motion_reduced"     = the EVALUATED video shows noticeably less object motion
                         than the BASE, but subjects still clearly move
  "motion_mostly_lost" = the EVALUATED video is near-static or frozen while the
                         BASE clearly moves - most of the object motion is gone
Judge motion only - ignore visual quality/aesthetics differences.
```

The reported motion score is $(\#\text{preserved} + \frac{1}{2}\,\#\text{reduced})/N \in [0, 1]$.

Joint motion+consistency (Consistency) judge prompt. The head-to-head verdict folds motion retention and consistency into a single call; the key design choice is that a frozen evaluated video is a *loss* even if it looks perfectly stable, which removes the static bias that a pure consistency rating would leave behind:

```
You are comparing two AI-generated videos made from the SAME text prompt.

Text prompt:
"""{prompt}"""

Two videos are attached: Video 1 (first) and Video 2 (second).
- The BASE model's output (the reference) is {base_label}.
- The video being EVALUATED is {eval_label}.

Decide whether the EVALUATED video ({eval_label}) is BETTER than the BASE
```

```
({base_label}). The EVALUATED video is better ONLY if it does BOTH of these,
relative to the BASE:
  (A) KEEPS THE MOTION - it preserves roughly as much genuine object/subject motion
      as the BASE (people, animals, vehicles, crowds, effects). "Motion" is real
      object/subject movement, NOT camera pans over a frozen scene and NOT texture
      flicker.
  (B) IS MORE CONSISTENT THAN THE BASE - on that moving content, objects hold their
      identity, shape, and structure BETTER than in the BASE: less morphing, warping,
      popping in/out, dissolving, or flicker than the BASE shows.

Being more consistent only counts if the motion is still there. A video that looks
cleaner only because it stopped moving has NOT kept the motion, so it is NOT better -
it is worse.

Answer with exactly one verdict (about the EVALUATED video relative to the BASE):
  "win"     = the EVALUATED video is BETTER than the BASE: it keeps about as much
              (or more) object motion AND is clearly more consistent than the BASE
              on that motion.
  "partial" = roughly a TIE: the EVALUATED video keeps the motion but is about as
              consistent as the BASE (no clear improvement either way) - neither
              clearly better nor clearly worse.
  "loss"    = the EVALUATED video is WORSE than the BASE: its motion is far behind
              the BASE (largely frozen / near-static), OR it is less consistent than
              the BASE (more morphing, identity changes, structural breakdown).
Judge motion and consistency only - ignore overall visual quality / aesthetics and
prompt wording. Consistency is a COMPARISON against the BASE, not an absolute; a
frozen or near-static EVALUATED video is always WORSE because it did not keep the
motion.
```

The reported win-rate is $(\#\text{win} + \frac{1}{2}\,\#\text{partial})/N$, centered at $50\,\%$; $>50\,\%$ means the method beats its own base.

Position debiasing. Both prompts refer to the videos through the *role* labels (BASE / EVALUATED) rather than the attachment slots, so the same question can be asked with the two videos in either physical order. Every pair is judged twice: once with the base attached first and once with the variant attached first, with {base_label} / {eval_label} swapped accordingly, so both verdicts mean "the variant relative to the base." Each of the two verdicts is mapped to the three-point numeric scale (Motion: mostly-lost$=0$, reduced$=\frac{1}{2}$, preserved$=1$; Consistency: loss$=0$, tie$=\frac{1}{2}$, win$=1$) and the pair's reported score is the *average* of its two order scores, so a video that wins in one order and ties in the other is credited 0.75 rather than being forced to one discrete verdict. Averaging the two symmetric presentations cancels the additive position bias exactly. Reported column values are the mean of these per-pair averages over the prompt set.

Reproducibility. The contested SF main-table pair (ours vs. World-R1) was run as three independent debiased passes; preserved% agrees within 1 pt across passes (ours 61.2–61.4, World-R1 63.4–64.4), and we report the final pass. Independent duplicate passes on two deployed rows differ by ≤ 0.005 motion score and ≤ 1.2 pp win-rate. Per-pair verdicts with per-order records and one-line reasons are archived as JSONL alongside the aggregates.

F Human Study Protocol

Design. A two-alternative forced-choice (2AFC) study on the LongLive backbone, which we prioritize because its 10.3 s horizon is where accumulated drift and the static-collapse shortcut matter most. We sample 50 prompts from the high-motion held-out subset and form three pairings per prompt and compare Stream4D against the distilled base, the World-R1 reproduction, and VideoGPA. Each comparison is judged exactly once; the 150 comparisons are partitioned across the 5 raters. For each pair the rater answers two questions: (1) which video shows *more natural motion*: enough real movement, not frozen; smooth, not jittery, and (2) which video is *better overall*: jointly weighing motion amount, motion quality, and object consistency.

Quality control. Each rater additionally sees two interleaved attention-check trials with an obviously frozen clip vs. a moving one; a rater is excluded if either check is failed. All 5 raters passed

Table S2: **Robustness check: main comparison on a uniformly random 500-prompt subset** of the same held-out VidProM set (seed-0 sample, no motion filter; 122 prompts overlap the high-motion subset of Table 1). Metrics and conventions as in Table 1 (*4DGT*: reconstructor-independent dynamic-reconstruction cross-check, Appendix D); all RL'd rows at ckpt-150, judged against the same backbone's base. ‡The VideoGPA row uses the standard VideoGPA recipe (the best-weight variant of Table 1 was generated only on the high-motion subset); on this unfiltered, low-motion-dominated subset its clean freeze sweeps *both* reconstructors. †marks methods whose Consistency win-rate falls below the 50 % break-even against their own base: on this low-motion-dominated subset, their reconstruction and VideoReward advantages are bought by suppressing motion (motion scores 0.14–0.48), not by generating better video. Bold = best RL'd row per backbone block; shaded = ours.

Method	MoVieS Recon.			4DGT Recon.			LLM judge		VideoReward win%			
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	Motion↑	Consist.↑	VQ↑	MQ↑	TA↑	Ovr↑
Distilled AR backbone: Self-Forcing (Huang et al., 2025) (5 s)												
Base (B0)	18.19	0.831	0.239	16.23	0.546	0.439	—	—	—	—	—	—
World-R1 (Wang et al., 2026a)	19.35	0.885	0.182	16.60	0.560	0.412	0.773	68.2	61.4	49.4	54.2	56.8
VideoGPA (Du et al., 2026)‡†	**24.88**	**0.908**	**0.143**	**19.87**	0.621	**0.396**	0.477	38.3	**63.0**	**63.4**	**65.0**	**67.4**
+ **Stream4D (Ours)**	21.23	0.895	0.177	17.70	**0.657**	0.411	**0.816**	**75.1**	59.8	56.6	54.4	59.8
Distilled AR backbone: Causal-Forcing (Zhu et al., 2026) (5 s)												
Base (B0)	16.30	0.802	0.252	14.94	0.512	0.455	—	—	—	—	—	—
World-R1 (Wang et al., 2026a)	19.67	**0.894**	0.162	16.34	0.575	0.417	0.637	61.1	**72.6**	60.6	**57.0**	65.8
VideoGPA (Du et al., 2026)‡†	**22.94**	0.843	**0.147**	**19.33**	0.605	**0.377**	0.484	36.5	67.6	54.8	53.4	62.2
+ **Stream4D (Ours)**	20.99	0.890	0.156	17.49	**0.650**	0.404	**0.716**	**64.9**	71.4	**65.8**	52.8	**68.6**
Distilled AR backbone: LongLive (Yang et al., 2025) (10.3 s)												
Base (B0)	18.11	0.863	0.217	15.45	0.509	0.447	—	—	—	—	—	—
World-R1 (Wang et al., 2026a)†	23.01	0.940	0.137	17.53	0.630	0.438	0.457	46.8	**74.8**	68.6	63.4	75.8
VideoGPA (Du et al., 2026)‡†	22.99	**0.943**	**0.130**	19.24	**0.668**	**0.339**	0.136	9.8	66.0	63.6	54.6	67.0
+ **Stream4D (Ours)**	**24.60**	0.910	0.147	**19.59**	0.646	0.417	**0.701**	**68.5**	**74.8**	**76.6**	**65.4**	**78.2**

On this unfiltered subset the frozen specialists sweep the reconstruction metrics under both *reconstructors—SF VideoGPA‡ takes the best MoVieS* and *4DGT PSNR (and the best VideoReward-Overall)—because a clean freeze of a near-static scene reconstructs well under any reconstructor. The judge is what exposes the shortcut: VideoGPA's motion score is* 0.477 *and it loses to its own base on Consistency (*38.3 %*), as do CF VideoGPA‡ (*36.5 %*), LL VideoGPA‡ (*9.8 %*), and LL World-R1 (*46.8 %*). Stream4D is the only method that beats its base under the joint verdict on all three backbones. Joint verdicts cover* 499/500 *prompts per row (one API failure each).*

both checks, so all 5 are retained and none are excluded; the reported numbers therefore use the full 50 judgments per comparison.

G Random-Subset Robustness Check

The main evaluation (Table 1) uses a motion-prominent filtered subset, chosen because it is the regime where the static-collapse shortcut matters. Table S2 repeats the comparison on a uniformly random 500-prompt subset of Vidprom, using the identical protocol. Because the pool is dominated by lower-motion prompts, the frozen VideoGPA and World-R1 sweep reconstruction and even VideoReward, yet under the motion-aware judge they lose to their own bases. Stream4D is the only method that beats its base under the joint verdict on all three backbones, and its judge scores are nearly unchanged from the motion-prominent subset (SF motion 0.816 vs 0.833; LL Consistency 68.5 vs 74.2).

H Additional Qualitative Results

Figure S1 shows motion-preservation filmstrips for one prompt per backbone (uniformly-spaced frames). Figures S2–S4 show the complementary failure, identical timestamps for every method: rollouts that move but deform, rollouts that buy stability by freezing, and Stream4D avoiding both.

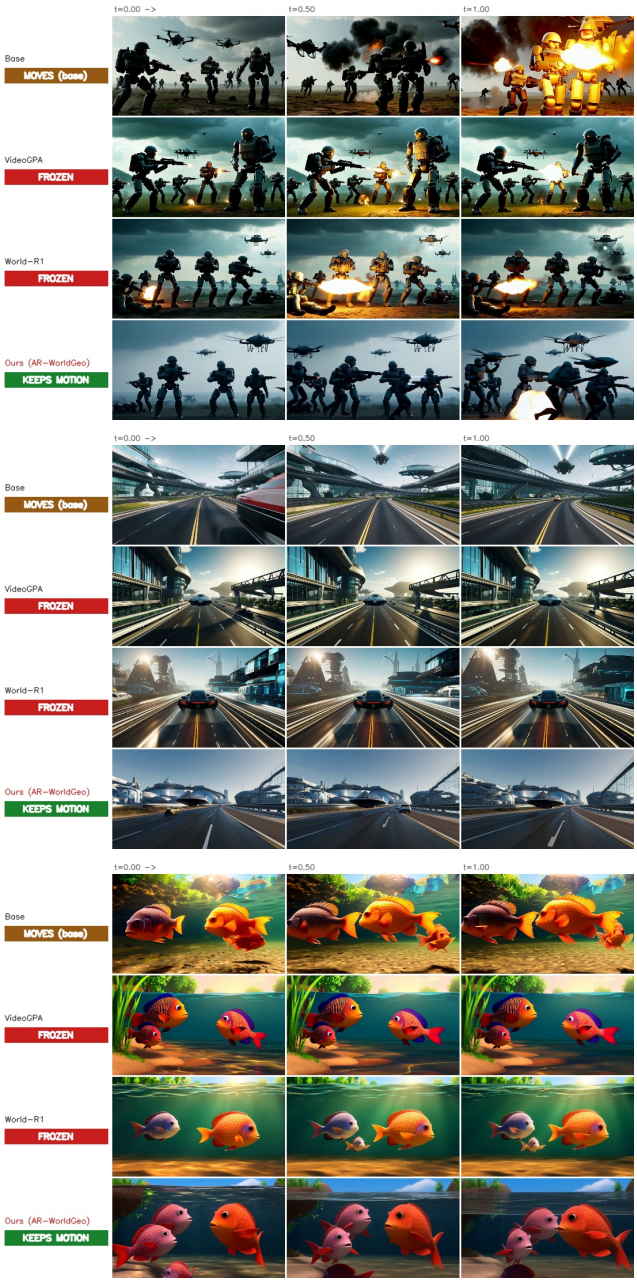


Figure S1: **Motion preserved vs. frozen: the static-reward shortcut, one prompt per backbone.** Top: Causal-Forcing (a WW3 battle); middle: Self-Forcing (a futuristic highway); bottom: LongLive (fish in a river). Three uniformly-spaced frames per method, all RL'd rows at ckpt-150; read each row *across* time. The static-3D baselines (VideoGPA, World-R1) are frozen—subjects hold the same pose from t=0 to t=1—while the distilled bases move but deform, and Stream4D keeps the motion while holding object identity and scene structure. Table 1 quantifies motion preservation over the full prompt set.



Figure S2: **Object-identity under motion: the base breaks the object apart, the static-reward baselines freeze the scene, and only ours keeps a *moving* object intact.**



Figure S3: **Consistency companion on Self-Forcing: an ancient-battlefield beast** Red outlines mark failure frames and the coloured badge names each row's behaviour. The failures are complementary: the *base* moves but **morphs**; the static-reward baselines are either **frozen**. Stream4D stays **coherent**: one beast, one identity, approaching the camera.



Figure S4: **Consistency companion on LongLive: a planet descending over a city** The base **deforms** into spreading debris/fire; VideoGPA's explosion **flickers** in and out over an otherwise static scene; World-R1 is **frozen**. Stream4D keeps a **coherent** descending planet over a stable city.