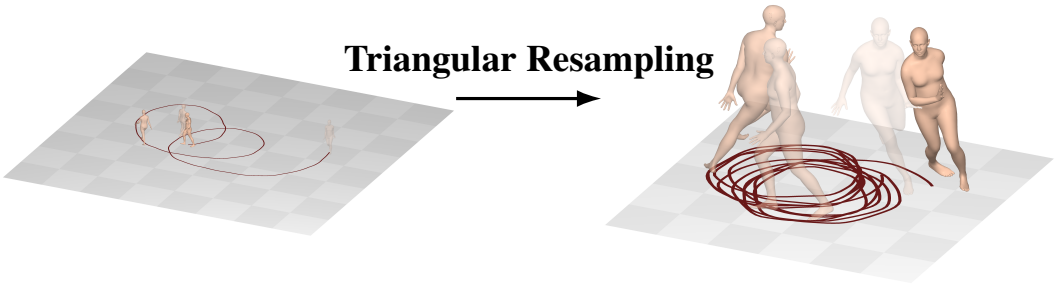
TRIANGULAR RESAMPLING FOR LONG-HORIZON MOTION GENERATION

Anonymous authors
Paper under double-blind review



Without TR: trajectory drift With TR: sustained circular motion

Prompt: "a person walks in a circle at a steady pace."

Figure 1: **Triangular Resampling (TR) improves long-horizon consistency.** Given the prompt "a person walks in a circle at a steady pace," the original model fails to sustain circular motion over time (left), while the same model fine-tuned with TR better maintains a circular trajectory (right). Red curves show the ground-plane root trajectories; earlier poses are more transparent and later poses are more opaque.

ABSTRACT

We introduce Triangular Resampling (TR), a post-training method for mitigating long-horizon error accumulation in motion diffusion models. TR builds on FloodDiffusion (Cai et al., 2026), which generates motion using a triangular denoising schedule. During training, the current model observes short, ground-truth-derived motion windows, whereas long-horizon inference repeatedly conditions on its own predictions, allowing errors to accumulate. A common remedy is to expose the model to its own rollouts during training (Zhao et al., 2025; Xiao et al., 2025). However, replacing only completed motion history does not capture the evolving, partially denoised states within a triangular denoising window. To address this mismatch, TR introduces a key design that extends rollout-based training to the entire active window, including partially denoised states. This exposes training to model-induced errors, but unrestricted rollout can also move the training states away from their paired ground-truth motion. We therefore introduce a denoising threshold to retain ground-truth anchoring while controlling the transition to model-generated rollouts. For each replayed training sample, TR draws a shared threshold and replays the multi-step triangular denoising trajectory without gradient tracking. After each update, states below the threshold are replaced with noise-matched ground truth, while states at or above it retain model predictions. The resulting latent window is then used in the standard forward pass, loss computation, and gradient update. This rollout construction supports both the original supervised objective (TR) and distribution matching (TR-DMD). We evaluate 120-second motion generation on HumanML3D test prompts. TR and TR-DMD achieve state-of-the-art FID AUC within their respective non-DMD and DMD comparison groups. Specifically, supervised TR reduces FID AUC by 40.9% and FID degradation slope by 55.3% relative to matched post-training without replay.

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.

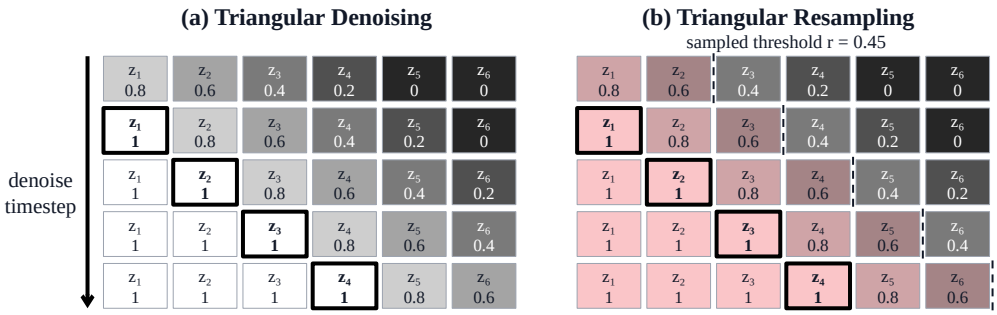


Figure 2: Comparison of triangular denoising and triangular resampling during training. The number in each cell is its denoising level α; light cells contain less noise and dark cells more. (a) Each pair of consecutive displayed rows represents two Euler updates; the intervening update is omitted for visual clarity. Each latent undergoes ten Euler updates in total, with $\Delta\alpha = 0.1$ per update and a 0.2 offset between adjacent latents; the thick border marks the newly committed latent. (b) TR replays the same trajectory with one shared per-sample threshold r (sampled as 0.45 here). For latent j, model rollout follows $\boldsymbol{z}_j^{k+1} = \boldsymbol{z}_j^k + (\alpha_j^{k+1} - \alpha_j^k) v_{\theta,j}(\boldsymbol{z}^k, \boldsymbol{\alpha}^k, \boldsymbol{c})$. After the update, TR sets $\tilde{\boldsymbol{z}}_j^{k+1} = \alpha_j^{k+1} \boldsymbol{z}_j^{\mathrm{GT}} + (1 - \alpha_j^{k+1}) \boldsymbol{\epsilon}_j^{k+1}$ when $\alpha_j^{k+1} < r$ (black-to-gray GT clamp), and $\tilde{\boldsymbol{z}}_j^{k+1} = \boldsymbol{z}_j^{k+1}$ when $\alpha_j^{k+1} \geq r$ (black-to-red model rollout), where $\boldsymbol{\epsilon}_j^{k+1} \sim \mathcal{N}(0, I)$. Replay repeats with the same threshold until the sampled time t.

1 Introduction

Long-horizon text-to-motion generation (Athanasiou et al., 2022; Shafir et al., 2024; Barquero et al., 2024; Lee et al., 2024) requires a model to sustain plausible movement and adherence to a text instruction far beyond the short sequences used for training. A central challenge is the mismatch between training on short, ground-truth-derived motion windows and repeatedly conditioning on model predictions during inference. Each prediction becomes part of the input for subsequent generation, allowing small errors to propagate and accumulate over time. Addressing this mismatch requires considering the states the model actually encounters throughout a long rollout.

We study this mismatch in FloodDiffusion (Cai et al., 2026), a state-of-the-art motion diffusion model with a triangular denoising schedule (Figure 2(a)). This representative schedule for sequential generation assigns increasing noise from earlier to later latents in the active window, reflecting the intuition that near-future motion is more constrained by the committed history. During inference, the leading token is committed when it becomes clean, while the remaining tokens continue denoising as the window advances. These partially denoised states retain predictions from earlier updates and also influence subsequent ones through bidirectional attention. Standard training, however, constructs window states directly by corrupting ground-truth latents to their prescribed noise levels, without the preceding model updates. Exposing the model to its own rollouts during training is a common remedy (Zhao et al., 2025; Xiao et al., 2025), but replacing only completed history leaves this mismatch unresolved in the partially denoised region.

We propose Triangular Resampling (TR), a post-training method that mitigates long-horizon error accumulation by addressing the train–inference mismatch in partially denoised states under triangular denoising. Its key design extends rollout-based training to partially denoised states while controlling their transition from ground-truth-derived states to model predictions (Figure 2(b)). Unrestricted rollout can move the training states away from their paired ground-truth motion. For each replayed sample, TR draws one denoising threshold, shared across all tokens and Euler updates, and replays the current model's multi-step triangular trajectory without gradient tracking. After each update, states below the threshold are replaced with noise-matched ground truth, while states at or above it retain model predictions. As denoising progresses, tokens cross the shared threshold and remain under model control. In this way, TR preserves a ground-truth reference to limit excessive drift during replay, while exposing training to model-induced errors to address the training–inference mismatch.

The rollout construction supports two training objectives. In supervised TR, the replayed latent window enters the standard training update, with targets derived from the paired clean motion. Alternatively, Distribution Matching Distillation (DMD) (Yin et al., 2024) aligns the distribution of generated samples with that of a pretrained diffusion teacher. We combine TR with DMD to obtain TR-DMD, adopting the DMD update recipe of Rolling Forcing (Liu et al., 2026) while retaining triangular replay.

We train on HumanML3D (Guo et al., 2022) and BABEL (Punnakkal et al., 2021), and evaluate 120-second motion generation on HumanML3D test prompts by computing metrics in twelve non-overlapping 10-second windows. Sharing the same backbone, initialization, and 30k post-training budget, TR reduces FID AUC from 1.951 to 1.153 (40.9%) and the FID linear degradation slope from 0.526 to 0.235 per minute (55.3%) compared with the baseline without TR (TR-off). Both TR and TR-DMD achieve state-of-the-art FID AUC within their respective non-DMD and DMD comparison groups under this protocol: 1.153 for TR and 1.324 for TR-DMD, compared with 1.351 for Rolling Forcing in the DMD group. A complementary pairwise preference evaluation ranks TR highest in both motion quality and text alignment, with Bradley–Terry scores of 0.302 and 0.317, respectively. The corresponding scores are -0.272 and -0.211 for TR-off, and -0.030 and -0.106 for Rolling Forcing.

Our contributions are:

- We identify a training–inference mismatch in the partially denoised states of triangular denoising and propose Triangular Resampling (TR) to mitigate the resulting long-horizon error accumulation. TR extends rollout-based training to these states, using a shared threshold to balance ground-truth anchoring and exposure to model predictions. This rollout construction supports both supervised training and distribution matching (TR-DMD).
- We systematically evaluate TR and TR-DMD for long-horizon motion generation. Both achieve state-of-the-art FID AUC within their respective non-DMD and DMD comparison groups under our evaluation protocol, with supervised TR improving over matched TR-off by 40.9%. Ablations examine the effects of clamping and replay frequency, while pairwise preference evaluation further supports TR's motion quality and text alignment.

2 Related Work

Long-Horizon Motion Generation. TEACH (Athanasiou et al., 2022) composes motion segments autoregressively, while DoubleTake (Shafir et al., 2024) refines overlapping segments from a short-motion diffusion prior. FlowMDM (Barquero et al., 2024) and T2LM (Lee et al., 2024) further study long-form motion synthesis through positional encoding and latent composition, respectively. FloodDiffusion (Cai et al., 2026) jointly denoises a bidirectional active window under a triangular schedule. Rather than proposing another composition or sampling architecture, we optimize the training-state distribution of this triangular generator for long-horizon motion.

Rollout Training in Motion Generation. DART (Zhao et al., 2025) trains on overlapping motion primitives with a staged curriculum that progresses from ground-truth histories through mixed histories to full diffusion rollouts. MotionStreamer (Xiao et al., 2025) uses Two-Forward training: an initial prediction pass supplies latents that replace a progressively increasing subset of history tokens before the standard training update. PRISM v1 (Ling et al., 2026) applies Self Forcing (Huang et al., 2025) to generated segments, decoding and re-encoding each segment as context for the next and optimizing the resulting rollouts through distribution matching. These approaches expose motion models to their own predictions through history replacement or self-conditioned rollout training. Our focus is the evolving joint state of a triangular denoising window, including both committed outputs and partially denoised tokens.

Rollout Training in Video Generation. Self Forcing (Huang et al., 2025) uses few-step autoregressive self-rollouts with truncated gradients and optimizes a video-level distribution-matching loss. Self Gradient Forcing (Zhuang et al., 2026) recomputes historical key–value representations in a parallel second pass, allowing future DMD (Yin et al., 2024) losses to train context encoding without differentiating through the serial rollout. Rolling Forcing (Liu et al., 2026) jointly denoises a rolling window at staggered noise levels, retains initial-frame attention sinks, and mixes rolling-window

and Self Forcing DMD updates with equal probability. Causal Forcing (Zhu et al., 2026) uses an autoregressive teacher for ODE initialization, followed by Self Forcing-style DMD. Resampling Forcing (Guo et al., 2025) instead resamples noise-corrupted ground-truth frames autoregressively with the current model, then conditions training on the detached resampled histories while retaining the original ground-truth-supervised flow-matching objective. It requires neither an auxiliary teacher nor a discriminator. Like Rolling Forcing, the FloodDiffusion backbone jointly denoises a window at staggered noise levels. Like Resampling Forcing, supervised TR uses detached model-generated states and retains ground-truth supervision. TR applies a shared denoising threshold throughout triangular replay. After each update, states below the threshold return to noise-matched ground truth, while states at or above it retain model predictions.

3 Preliminaries: FloodDiffusion Triangular Denoising

Let $\boldsymbol{z}_{1:T}^{\mathrm{GT}}$ be clean motion latents and $\boldsymbol{c}$ the text condition. FloodDiffusion (Cai et al., 2026) uses a linear flow-matching path (Lipman et al., 2023) with a triangular denoising schedule. Let τ denote its continuous global denoising phase. The clean-data coefficient of latent position j is

$$\alpha_j(\tau) = \mathrm{clip}\left(\tau - \frac{j}{c}, 0, 1\right), \tag{1}$$

where c controls the slope and the number of simultaneously active tokens. The clean and noisy boundaries are

$$m(\tau) = \lceil (\tau - 1)c \rceil, \qquad n(\tau) = \lceil \tau c \rceil, \tag{2}$$

so positions before $m(\tau)$ are clean, positions from $m(\tau)$ to $n(\tau) - 1$ form the active denoising window, and later positions remain noise. This produces the triangular schedule in Figure 2(a).

At inference, let k denote the Euler integration step and τ_k the corresponding global denoising phase. The solver advances

$$\tau_{k+1} = \tau_k + \Delta\tau, \qquad \Delta\tau = \frac{1}{N}, \qquad \alpha_j^k := \alpha_j(\tau_k), \tag{3}$$

where N is the number of Euler steps per unit phase. Each step therefore advances the triangular frontier by $c\Delta\tau = c/N$ positions. Token j reaches the clean endpoint at $\tau = 1 + j/c$, so successive commits are separated by $1/c$ in phase, or N/c Euler steps when N is divisible by c. Our configuration uses $c = 5$ and $N = 10$: one token is committed every two Euler steps after the initial warm-up. A standard training state at a sampled phase τ is constructed directly from data,

$$\boldsymbol{z}_j(\tau) = \alpha_j(\tau)\boldsymbol{z}_j^{\mathrm{GT}} + (1 - \alpha_j(\tau))\boldsymbol{\epsilon}_j, \qquad \boldsymbol{\epsilon}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{4}$$

and a bidirectional denoiser $v_\theta(\boldsymbol{z}(\tau), \boldsymbol{\alpha}(\tau), \boldsymbol{c})$ predicts velocities for the active tokens. An Euler update advances each position according to its actual change in denoising level,

$$\boldsymbol{z}_j^{k+1} = \boldsymbol{z}_j^k + (\alpha_j^{k+1} - \alpha_j^k) v_{\theta,j}(\boldsymbol{z}^k, \boldsymbol{\alpha}^k, \boldsymbol{c}). \tag{5}$$

Once the leading token becomes clean, it is committed, the window shifts by one position, and fresh noise enters at the trailing edge. Repeating these operations extends generation beyond the training sequence length while retaining a bounded active window.

4 Triangular Resampling

Standard FloodDiffusion training constructs each active window from ground-truth motion along the forward path in Equation 4, whereas inference repeatedly updates, commits, and shifts model-generated states. The model therefore trains on ground-truth-derived windows but conditions on accumulated prediction errors during long rollouts.

This mismatch affects both completed history and partially denoised states within the active window. We propose Triangular Resampling to replay both regions (Figure 2), with GT clamp controlling their exposure to model errors. Below-threshold states follow the noise-matched ground-truth path; states reaching the threshold retain model predictions. Replay construction, threshold control, and optimization are separate components: the supervised variant retains the original clean-motion target, while TR-DMD applies distribution matching to clean predictions read out from the replay.

4.1 GT-CLAMPED TRIANGULAR ROLLOUT

Each post-training sample with a nonempty history enters replay with probability γ, the *resampling ratio*. For a replayed sample, we select an interval ending at the current output band and beginning a bounded number of tokens earlier, reset this interval to Gaussian noise at zero denoising progress, and keep any earlier prefix clean. The current model then follows the native triangular schedule through successive Euler updates with phase increment $\Delta\tau = 1/N$ until the sampled training phase is reached, including the corresponding commits, window shifts, and fresh-noise injections. The resulting replay state is detached and passed to the subsequent optimization step.

For each replayed sample, we draw one release threshold using the shifted logit-normal parameterization (Guo et al., 2025):

$$r = \mathrm{sigmoid}(u + \log s), \qquad u \sim \mathcal{N}(0, 1), \tag{6}$$

where the shift $s > 0$ controls clamp strength. The sigmoid maps an unconstrained Gaussian draw to a valid denoising threshold in $(0, 1)$, while $\log s$ shifts its log-odds. This provides smooth control over the release point: smaller s favors earlier release to model rollout, whereas larger s retains ground-truth anchoring longer. The same r is shared by all tokens and Euler steps in that replay. After each Euler update, token j is replaced according to

$$\widetilde{\boldsymbol{z}}_j^{k+1} = \begin{cases} \alpha_j^{k+1}\boldsymbol{z}_j^{\mathrm{GT}} + (1-\alpha_j^{k+1})\boldsymbol{\epsilon}_j^{k+1}, & \alpha_j^{k+1} < r, \\ \boldsymbol{z}_j^{k+1}, & \alpha_j^{k+1} \geq r, \end{cases} \tag{7}$$

with fresh $\boldsymbol{\epsilon}_j^{k+1} \sim \mathcal{N}(0, I)$ for the clamped branch. The clamp resamples the ground-truth latent at the location's current triangular noise level. As denoising progresses, a token crosses the common frontier and remains under model control. Sharing the threshold preserves a contiguous denoising frontier across the window. The post-clamp state $\widetilde{\boldsymbol{z}}^{k+1}$ becomes the input to the next Euler update.

The two hyperparameters have distinct roles. The ratio γ determines how often optimization sees a replayed sample. The shift s determines how much of an entered replay is anchored to ground truth. Small s releases tokens early and approaches free rollout; large s keeps more states clamped. TR-off is defined by $\gamma = 0$. A fixed s specifies a fixed threshold distribution.

4.2 OPTIMIZATION OBJECTIVES

The replay construction specifies the training states, independently of the objective applied to model predictions. We use it with clean-motion supervision in TR and with distribution matching in TR-DMD.

GT-Supervised TR. For a training example sampled at phase τ, let $\widetilde{\boldsymbol{z}}$ denote the detached latent sequence left by the triangular replay of Section 4.1, and let $\mathcal{B}(\tau)$ denote the output band supervised at this step. Replay ends at phase τ, so the entries of $\widetilde{\boldsymbol{z}}$ sit at the denoising levels $\boldsymbol{\alpha}(\tau)$ prescribed by Equation 1. The optimization forward pass directly takes this replayed state as input, retaining its model-induced errors. For the linear path, we recover an effective noise from the replay state and the paired clean latent,

$$\widehat{\boldsymbol{\epsilon}}_j = \frac{\widetilde{\boldsymbol{z}}_j - \alpha_j(\tau)\boldsymbol{z}_j^{\mathrm{GT}}}{\max(1-\alpha_j(\tau), \epsilon)}, \qquad \boldsymbol{v}_j^\star = \boldsymbol{z}_j^{\mathrm{GT}} - \widehat{\boldsymbol{\epsilon}}_j, \tag{8}$$

where ϵ is a small numerical floor. When $1 - \alpha_j(\tau) \geq \epsilon$, Equation 8 recovers the sampled noise for clamped tokens, giving the standard flow-matching target. Under the same condition, it defines the velocity that carries a model-generated state to the paired clean latent. The loss is

$$\mathcal{L}_{\mathrm{TR}}(\theta) = \mathbb{E}\left[\frac{1}{|\mathcal{B}(\tau)|}\sum_{j\in\mathcal{B}(\tau)} \left\| v_{\theta,j}(\widetilde{\boldsymbol{z}}, \boldsymbol{\alpha}(\tau), \boldsymbol{c}) - \boldsymbol{v}_j^\star \right\|_2^2\right]. \tag{9}$$

Supervision is applied to $\mathcal{B}(\tau)$, while the remaining tokens provide context; the entire replayed window is treated as a fixed input. The objective therefore teaches the model how to continue denoising toward the paired clean motion when its own rollout displaces the active window. Samples that bypass replay use the ordinary forward-path state of Equation 4, for which Equation 9 recovers the original FloodDiffusion objective.

Distribution Matching with TR-DMD. TR-DMD changes the optimization target while retaining triangular Euler replay and GT clamp. Propagated replay states remain detached. For each sample, we select one discrete denoising stage uniformly from the updates at which the threshold permits model control. When latent j reaches that stage, its input state is saved and its velocity is recomputed with gradients, yielding the clean prediction

$$\widehat{\boldsymbol{z}}_{0,j} = \mathrm{sg}(\boldsymbol{z}_{\alpha_j,j}) + (1-\alpha_j) v_{\theta,j}\big(\mathrm{sg}(\boldsymbol{z}_{\boldsymbol{\alpha}}), \boldsymbol{\alpha}, \boldsymbol{c}\big), \tag{10}$$

where sg stops gradients. The recomputed clean predictions form the samples used for distribution matching. We optimize these predictions with a sampled-stage truncated-gradient approximation, treating the replayed states as fixed inputs.

Following the DMD update recipe of Rolling Forcing (Liu et al., 2026), a frozen real-score model and a trainable fake-score model define a normalized, detached distribution-matching direction for the generator. The fake score learns a flow-matching objective on newly sampled, detached generator outputs. Both scores independently initialize from the same motion-teacher EMA; the generator initializes from a supervised TR checkpoint. GT anchors the replay states, while DMD supplies the generator's training objective. For distribution matching, we use up to the final 35 valid latents of each generated sequence (Appendix A).

5 Experiments

We organize the evaluation around four research questions. RQ1: Does controlled triangular rollout improve long-horizon motion quality and text alignment (Section 5.2)? RQ2: How do TR and TR-DMD compare with existing methods within their respective objective groups (Section 5.2)? RQ3: How do clamp strength, release structure, replay ratio, and threshold curriculum affect generation (Section 5.3)? RQ4: How do blinded pairwise video preferences complement the quantitative metrics (Section 5.4)?

5.1 Experimental Settings

Datasets. We train on HumanML3D (Guo et al., 2022) and BABEL (Punnakkal et al., 2021). Both datasets use the standard 263-dimensional HumanML3D motion representation at 20 fps. Evaluation uses prompts from the official HumanML3D test split.

Evaluation Protocol. Prior long-horizon motion evaluation assesses action segments and transitions (Shafir et al., 2024; Barquero et al., 2024), or uses sliding windows to evaluate extended sequences (Lee et al., 2024). We focus on how motion quality and text alignment evolve during sustained generation under a fixed text instruction. We generate 120-second sequences for 256 frozen HumanML3D test prompts and divide each sequence into twelve non-overlapping 10-second windows. At each window position, we compute FID, Matching Distance, and R-precision using the standard HumanML3D evaluator (Guo et al., 2022). We report the resulting temporal curves, their normalized area under the curve (AUC), and linear degradation slopes. AUC summarizes performance across the full horizon, while slopes capture its temporal trend. We use FID AUC as the primary criterion for overall generation quality; the remaining metrics provide complementary evidence on text alignment and temporal degradation.

Training Setup. We initialize motion generators from the official FloodDiffusion checkpoint and use its corresponding pretrained VAE. TR-off and TR post-train for 30,000 optimization steps with the same data order, optimizer, learning-rate schedule, batch size, and single training seed. Each supervised post-training run uses one NVIDIA H200 GPU. Cumulative run times are approximately 21 hours for TR-off and 29 hours for the main TR configuration ($\gamma = 0.25$, $s = 0.6$). TR-DMD initializes its generator from the 30k supervised TR checkpoint at $(\gamma, s) = (1, 0.6)$ and retains these replay settings during distribution matching. Its additional distribution-matching stage runs on two NVIDIA H200 GPUs for approximately 13 hours, performing 1,200 outer iterations with 1,200 fake-score updates and 240 generator updates. Additional implementation details are provided in Appendix A.

Baselines. We select baselines to cover complementary strategies for addressing train–inference mismatch. TR-off and Gaussian history noise provide same-backbone controls for post-training

and generic history corruption. DART (Zhao et al., 2025) and MotionStreamer (Xiao et al., 2025) represent motion-domain training with model-induced histories. DART generates motion primitives, while MotionStreamer generates individual latents. Resampling Forcing (Guo et al., 2025) provides a teacher-free, supervised self-resampling alternative. Self Forcing (Huang et al., 2025), Self Gradient Forcing (Zhuang et al., 2026), Rolling Forcing (Liu et al., 2026), and Causal Forcing (Zhu et al., 2026) initialization followed by Rolling Forcing provide video-derived DMD baselines. These video-derived methods use multi-latent chunks as autoregressive units. In contrast, our FloodDiffusion backbone jointly denoises an active window at staggered noise levels and commits one latent at a time. Adaptation protocols and training budgets are detailed in Appendix B.

5.2 Main Results

Table 1 reports the TR configuration selected by FID AUC on the same evaluation set. It improves over TR-off with the same backbone and number of post-training updates. FID AUC decreases from 1.951 to 1.153 (40.9%), and the FID degradation slope falls from 0.526 to 0.235 per minute. Matching Distance AUC improves from 3.864 to 3.786, while R-precision AUC increases from 0.648 to 0.655. Although TR-DMD improves FID AUC over TR-off, it does not improve on its supervised $(\gamma, s) = (1, 0.6)$ initialization (1.251 in Table 2); compatibility with DMD therefore does not imply an additional fidelity gain over supervised TR. Figure 3 shows how the supervised TR–TR-off differences evolve over time.

Table 1: Main comparisons grouped by without or with DMD. Following HumanML3D (Guo et al., 2022), entries report means with 95% confidence intervals after 20 rounds of evaluations. Slopes are per minute, with R-precision slopes in percentage points.

Method	FID		Matching Distance		R-precision	
	AUC↓	slope↓	AUC↓	slope↓	AUC↑	slope↑
Without DMD						
TR ($\gamma = 0.25$, $s = 0.6$)	$\mathbf{1.153^{\pm 0.030}}$	$0.235^{\pm 0.035}$	$3.786^{\pm 0.013}$	$0.334^{\pm 0.015}$	$0.655^{\pm 0.005}$	$-5.804^{\pm 0.485}$
TR-off (Cai et al., 2026)	$1.951^{\pm 0.068}$	$0.526^{\pm 0.070}$	$3.864^{\pm 0.026}$	$0.399^{\pm 0.028}$	$0.648^{\pm 0.007}$	$-6.837^{\pm 0.574}$
Gaussian noise ($\sigma = 0.05$)	$\underline{1.410}^{\pm 0.035}$	$0.306^{\pm 0.052}$	$3.847^{\pm 0.020}$	$0.316^{\pm 0.030}$	$0.656^{\pm 0.006}$	$-5.663^{\pm 0.597}$
DART (Zhao et al., 2025)	$13.192^{\pm 0.134}$	$-2.687^{\pm 0.266}$	$5.300^{\pm 0.003}$	$-0.080^{\pm 0.023}$	$0.418^{\pm 0.004}$	$1.276^{\pm 1.154}$
MotionStreamer (Xiao et al., 2025)	$7.376^{\pm 0.436}$	$3.363^{\pm 0.064}$	$5.771^{\pm 0.097}$	$0.812^{\pm 0.027}$	$0.385^{\pm 0.011}$	$-11.533^{\pm 0.990}$
Resampling Forcing (Guo et al., 2025)	$3.407^{\pm 0.140}$	$1.573^{\pm 0.154}$	$3.929^{\pm 0.008}$	$0.475^{\pm 0.037}$	$0.650^{\pm 0.009}$	$-7.307^{\pm 1.457}$
With DMD						
TR-DMD ($\gamma = 1$, $s = 0.6$)	$\mathbf{1.324^{\pm 0.051}}$	$0.442^{\pm 0.068}$	$3.785^{\pm 0.018}$	$0.270^{\pm 0.006}$	$0.633^{\pm 0.012}$	$-4.750^{\pm 0.282}$
Self Forcing (Huang et al., 2025)	$1.520^{\pm 0.086}$	$-0.020^{\pm 0.118}$	$3.566^{\pm 0.009}$	$0.040^{\pm 0.053}$	$0.690^{\pm 0.019}$	$-0.284^{\pm 1.083}$
Self Gradient Forcing (Zhuang et al., 2026)	$4.005^{\pm 0.435}$	$0.322^{\pm 0.279}$	$4.507^{\pm 0.123}$	$0.282^{\pm 0.094}$	$0.541^{\pm 0.028}$	$-3.182^{\pm 2.008}$
Rolling Forcing (Liu et al., 2026)	$\underline{1.351}^{\pm 0.061}$	$0.026^{\pm 0.097}$	$3.574^{\pm 0.018}$	$-0.010^{\pm 0.037}$	$0.694^{\pm 0.017}$	$0.270^{\pm 0.454}$
Causal Forcing Init. (Zhu et al., 2026) + Rolling Forcing	$1.634^{\pm 0.035}$	$0.116^{\pm 0.102}$	$3.199^{\pm 0.033}$	$0.000^{\pm 0.055}$	$0.749^{\pm 0.007}$	$-0.268^{\pm 0.600}$

Without DMD, TR achieves the best FID AUC (1.153). Even TR-off already outperforms DART, MotionStreamer, and Resampling Forcing in both FID AUC (1.951 versus 13.192, 7.376, and 3.407) and Matching Distance AUC (3.864 versus 5.300, 5.771, and 3.929). This strong baseline supports the effectiveness of FloodDiffusion's triangular, latent-wise denoising strategy for long-horizon generation under our protocol. Gaussian history noise further reduces FID AUC to 1.410 and Matching Distance AUC to 3.847, making it a competitive same-backbone baseline. Nevertheless, TR improves these scores to 1.153 and 3.786, respectively, including an 18.2% reduction in FID AUC relative to Gaussian noise. Structured triangular replay therefore yields greater fidelity gains than generic history corruption in this controlled comparison.

With DMD, TR-DMD achieves the best FID AUC (1.324), followed by Rolling Forcing (1.351), Self Forcing (1.520), Causal Forcing initialization followed by Rolling Forcing (1.634), and Self Gradient Forcing (4.005). Rolling Forcing (Liu et al., 2026) likewise uses staggered noise levels, but operates on multi-latent chunks rather than committing individual latents. Its second-place result, together with the strong TR results, is consistent with the benefit of staggered denoising schedules for long-horizon motion fidelity.

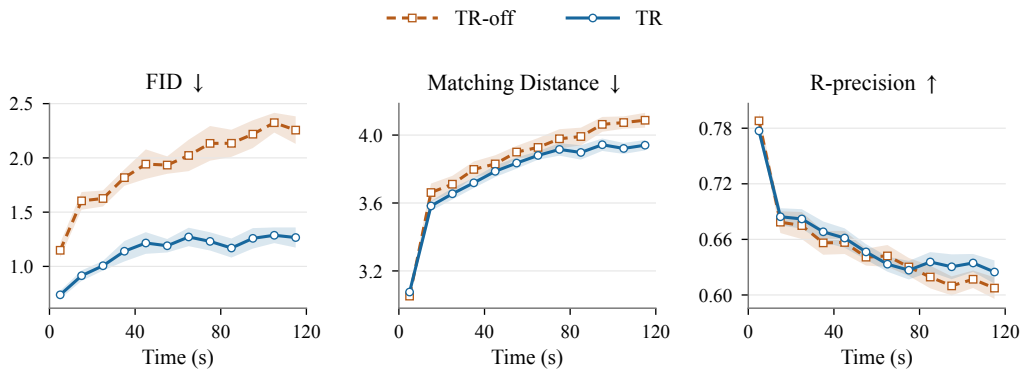


Figure 3: Window-wise quality and text alignment for TR-off and TR with $(\gamma, s) = (0.25, 0.6)$. Each point summarizes a non-overlapping 10-second window over 256 prompts. Curves show means and shaded 95% confidence intervals, under the convention in Table 1. Lower FID and Matching Distance and higher R-precision indicate better performance.

5.3 Ablation Study

Table 2: Triangular Resampling ablations. Entries follow the mean and 95% confidence-interval convention of Table 1. Bold rows mark the lowest FID AUC within each group.

Group	Setting	FID		Matching Distance		R-precision	
		AUC↓	slope↓	AUC↓	slope↓	AUC↑	slope↑
Shift s ($\gamma = 1$)	0 (full self-rollout)	$16.024^{\pm 0.064}$	$1.151^{\pm 0.065}$	$6.189^{\pm 0.006}$	$0.240^{\pm 0.008}$	$0.357^{\pm 0.004}$	$-3.935^{\pm 0.408}$
	0.3	$2.044^{\pm 0.047}$	$0.459^{\pm 0.062}$	$3.853^{\pm 0.020}$	$0.285^{\pm 0.014}$	$0.631^{\pm 0.005}$	$-4.335^{\pm 0.354}$
	0.6	$\mathbf{1.251^{\pm 0.024}}$	$\mathbf{0.376^{\pm 0.039}}$	$\mathbf{3.780^{\pm 0.014}}$	$\mathbf{0.259^{\pm 0.021}}$	$\mathbf{0.646^{\pm 0.005}}$	$\mathbf{-4.391^{\pm 0.492}}$
	1	$1.924^{\pm 0.054}$	$0.650^{\pm 0.063}$	$4.020^{\pm 0.019}$	$0.417^{\pm 0.018}$	$0.639^{\pm 0.006}$	$-6.103^{\pm 0.403}$
	3	$1.349^{\pm 0.036}$	$0.300^{\pm 0.060}$	$3.663^{\pm 0.011}$	$0.287^{\pm 0.029}$	$0.679^{\pm 0.004}$	$-5.041^{\pm 0.335}$
	10	$1.983^{\pm 0.063}$	$0.425^{\pm 0.084}$	$3.659^{\pm 0.018}$	$0.240^{\pm 0.031}$	$0.681^{\pm 0.006}$	$-4.036^{\pm 0.391}$
	100	$2.765^{\pm 0.072}$	$0.967^{\pm 0.089}$	$3.819^{\pm 0.018}$	$0.354^{\pm 0.025}$	$0.654^{\pm 0.005}$	$-5.837^{\pm 0.627}$
	1000	$2.780^{\pm 0.095}$	$1.004^{\pm 0.158}$	$3.832^{\pm 0.023}$	$0.375^{\pm 0.036}$	$0.654^{\pm 0.005}$	$-6.067^{\pm 0.638}$
	10000	$2.780^{\pm 0.095}$	$1.004^{\pm 0.158}$	$3.832^{\pm 0.023}$	$0.375^{\pm 0.036}$	$0.654^{\pm 0.005}$	$-6.067^{\pm 0.638}$
	Random-position clamp	$1.857^{\pm 0.052}$	$0.510^{\pm 0.052}$	$3.841^{\pm 0.018}$	$0.374^{\pm 0.023}$	$0.643^{\pm 0.004}$	$-6.255^{\pm 0.602}$
	Curriculum ($3 \to 0.6$)	$1.455^{\pm 0.015}$	$0.348^{\pm 0.025}$	$3.877^{\pm 0.034}$	$0.287^{\pm 0.020}$	$0.638^{\pm 0.010}$	$-5.100^{\pm 0.641}$
Ratio γ ($s = 0.6$)	**0.25**	$\mathbf{1.153^{\pm 0.030}}$	$\mathbf{0.235^{\pm 0.035}}$	$\mathbf{3.786^{\pm 0.013}}$	$\mathbf{0.334^{\pm 0.015}}$	$\mathbf{0.655^{\pm 0.005}}$	$\mathbf{-5.804^{\pm 0.485}}$
	0.5	$1.495^{\pm 0.023}$	$0.438^{\pm 0.049}$	$3.929^{\pm 0.014}$	$0.349^{\pm 0.024}$	$0.622^{\pm 0.004}$	$-6.038^{\pm 0.486}$
	1	$1.251^{\pm 0.024}$	$0.376^{\pm 0.039}$	$3.780^{\pm 0.014}$	$0.259^{\pm 0.021}$	$0.646^{\pm 0.005}$	$-4.391^{\pm 0.492}$

All ablations start from the official pretrained FloodDiffusion checkpoint and use 30k post-training updates. Table 2 varies the threshold shift s at $\gamma = 1$ and the replay probability γ at $s = 0.6$, with contiguous-frontier clamping by default. Random-position clamping uses $(\gamma, s) = (1, 0.6)$ and uniformly redistributes the frontier mask's exact clamp count over the replay region at each Euler step. Curriculum fixes $\gamma = 1$, cosine-interpolates $\log s$ from $\log 3$ to $\log 0.6$ over the first 15k updates, and holds $s = 0.6$ for the remaining 15k. Full self-rollout, labeled 0, disables GT clamping rather than setting $s = 0$.

For threshold control, full self-rollout severely degrades motion quality despite the pretrained initialization: its FID AUC reaches 16.024, compared with 1.251 for GT-clamped replay at $s = 0.6$. R-precision AUC also falls from 0.646 to 0.357, demonstrating the importance of GT anchoring in this setting. The shift sweep identifies an empirical optimum at $s = 0.6$ among the tested values: FID AUC is 2.044 at $s = 0.3$, 1.349 at $s = 3$, and at least 2.765 for $s \geq 100$. For $s = 1000$ and $s = 10000$, thresholds concentrate near 1, keeping nearly all intermediate denoising states clamped to noise-matched GT. Both settings thus approach GT-based training and produce identical evaluation

results in our runs. Random-position clamping retains much of the benefit of GT anchoring, reaching FID AUC 1.857, while curriculum performs better at 1.455. Neither matches the fixed $s = 0.6$ threshold distribution (1.251), indicating that a well-chosen fixed shift with contiguous release is sufficient for the best observed fidelity in this sweep.

For replay probability, the tested nonzero ratios show a non-monotonic pattern: both infrequent replay ($\gamma = 0.25$) and replay on every eligible sample ($\gamma = 1$) outperform the intermediate mixture ($\gamma = 0.5$). Their FID AUCs are 1.153, 1.251, and 1.495, respectively, with corresponding degradation slopes of 0.235, 0.376, and 0.438. Thus, within this grid, either lower or higher replay frequency is preferable to an equal mixture of replayed and ordinary training samples; $\gamma = 0.25$ gives the best FID AUC and slope.

5.4 Human Evaluation

To assess the overall visual quality of generated motion, we conduct a pairwise preference test. We compare TR, the matched TR-off (FloodDiffusion) baseline, and Rolling Forcing using 30 prompts, with one generated motion per method and prompt. For each prompt, all three method pairs are presented side by side in three rounds, with method identities hidden and left–right order varied across rounds. Each comparison yields separate choices for motion quality and text alignment. We aggregate all 270 pairwise choices per criterion into Bradley–Terry (BT) scores. Appendix C provides the protocol and complete prompt list.

Table 3 shows that TR achieves the highest BT scores for both motion quality and text alignment. For motion quality, TR scores 0.302, compared with -0.272 for TR-off and -0.030 for Rolling Forcing. For text alignment, the corresponding scores are 0.317, -0.211, and -0.106. Both criteria yield the same ranking, with a larger score gap between TR and TR-off than between TR and Rolling Forcing. This consistent ranking indicates that TR is preferred for both motion quality and text alignment among the compared methods.

Table 3: Pairwise video preferences on 30 prompts. Three rounds provide three judgments per pair and prompt (270 comparisons), with separate choices for motion quality and text alignment. Entries are zero-mean log-strength Bradley–Terry scores fitted jointly across all three rounds; higher is better. Bold marks the highest score in each column.

Method	Motion Quality ↑	Text Alignment ↑
TR	**0.302**	**0.317**
TR-off (Cai et al., 2026)	−0.272	−0.211
Rolling Forcing (Liu et al., 2026)	−0.030	−0.106

6 Conclusions

We propose Triangular Resampling (TR) to improve long-horizon motion generation under triangular denoising. TR replays the active-window trajectory, including partially denoised states, and uses ground-truth clamping to control their release into model rollouts. Supervised TR reduces FID AUC by 40.9% relative to matched TR-off. TR and TR-DMD attain the lowest mean FID AUC within their respective comparison groups under this protocol. Ablations support controlled GT anchoring over full self-rollout or random-position clamping, while TR-DMD demonstrates compatibility with distribution matching. These results highlight the importance of train-inference match in the active denoising area beyond the completed history. We leave extensions to other backbones, denoising schedules, and post-training data settings to future work.

Limitations. Multi-step replay slows post-training and increases its computational cost. In our initial 5k-update runs on shared GPUs, TR at $\gamma = 0.25$ takes approximately 1.95 seconds per update, compared with 0.58 seconds for TR-off (about $3.3\times$). Shorter stochastic rollouts combined with sampled-step supervision and gradient truncation, inspired by Self Forcing (Huang et al., 2025), may help reduce this overhead.

Ethics Statement

We use existing motion datasets and pretrained models in accordance with their respective licenses and access terms. Our human evaluation respects participants' rights, dignity, and privacy. Participation is voluntary and based on informed consent, and participants can withdraw at any time. Participants receive compensation at an effective hourly rate above the local average hourly wage. All responses are de-identified and used solely for research purposes.

Reproducibility Statement

Equations 1–10 specify the schedule, replay, clamp distribution, supervised target, and differentiable DMD readout. Section 5 records the evaluation horizon, windowing, supervised and DMD update budgets, initialization, and ablation grid. Appendix B records the external-method training and inference protocols. We will publicly release the implementation, training configurations, frozen prompt manifests, checkpoint identifiers, and evaluation scripts to support reproducibility.

AI Use Statement

Generative AI tools were used to assist with literature discovery and summarization, experiment-planning feedback, software implementation and debugging, and drafting and editing portions of this manuscript and bibliography. The authors reviewed the cited primary sources, technical descriptions, code behavior, and all AI-assisted text, and take responsibility for the final content, claims, and artifacts of the submission.

References

Nikos Athanasiou, Mathis Petrovich, Michael J. Black, and Gül Varol. TEACH: Temporal action composition for 3d humans. In *International Conference on 3D Vision*, pp. 414–423, 2022.

German Barquero, Sergio Escalera, and Cristina Palmero. Seamless human motion composition with blended positional encodings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 457–469, 2024.

Yiyi Cai, Yuhan Wu, Kunhang Li, You Zhou, Bo Zheng, and Haiyang Liu. FloodDiffusion: Tailored diffusion forcing for streaming motion generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2295–2304, 2026.

Chuan Guo, Shihao Zou, Xinxin Zuo, Sen Wang, Wei Ji, Xingyu Li, and Li Cheng. Generating diverse and natural 3d human motions from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5152–5161, 2022.

Yuwei Guo, Ceyuan Yang, Hao He, Yang Zhao, Meng Wei, Zhenheng Yang, Weilin Huang, and Dahua Lin. End-to-end training for autoregressive video diffusion via self-resampling. *arXiv preprint arXiv:2512.15702*, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, volume 38, pp. 167283–167308, 2025. doi: 10.52202/085713-5576.

Taeryung Lee, Fabien Baradel, Thomas Lucas, Kyoung Mu Lee, and Grègory Rogez. T2LM: Long-term 3d human motion generation from multiple sentences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1867–1876, 2024.

Zeyu Ling, Qing Shuai, Teng Zhang, Shiyang Li, Bo Han, and Changqing Zou. PRISM: Streaming human motion generation with per-joint latent decomposition. *arXiv preprint arXiv:2603.08590v1*, 2026. Version 1.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *International Conference on Learning Representations*, pp. 91177–91196, 2026.

Abhinanda R. Punnakkal, Arjun Chandrasekaran, Nikos Athanasiou, Alejandra Quiros-Ramirez, and Michael J. Black. BABEL: Bodies, action and behavior with english labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 722–731, 2021.

Yonatan Shafir, Guy Tevet, Roy Kapon, and Amit H. Bermano. Human motion diffusion as a generative prior. In *International Conference on Learning Representations*, pp. 8717–8733, 2024.

Lixing Xiao, Shunlin Lu, Huaijin Pi, Ke Fan, Liang Pan, Yueer Zhou, Ziyong Feng, Xiaowei Zhou, Sida Peng, and Jingbo Wang. MotionStreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10086–10096, 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo 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.

Kaifeng Zhao, Gen Li, and Siyu Tang. DartControl: A diffusion-based autoregressive motion model for real-time text-driven motion control. In *International Conference on Learning Representations*, pp. 23569–23592, 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 *Proceedings of the International Conference on Machine Learning*, 2026.

Junhao Zhuang, Shiyi Zhang, Yuxuan Bian, Yaowei Li, Yawen Luo, Yijun Liu, Weiyang Jin, Songchun Zhang, Xianglong He, Xuying Zhang, Haoran Li, Haoyang Huang, Zeyue Xue, and Nan Duan. Self gradient forcing: Native long video extrapolation. *arXiv preprint arXiv:2607.20368*, 2026.

A TR AND TR-DMD IMPLEMENTATION DETAILS

Supervised TR. Checkpoints are saved every 5,000 optimization steps, and formal comparisons use the final 30k checkpoint. The main supervised TR uses $(\gamma, s) = (0.25, 0.6)$ and contiguous-frontier clamping, the lowest-FID-AUC configuration in the reported ratio sweep (Table 2). This hyperparameter choice uses the reported evaluations, not a held-out selection set.

TR-DMD Optimization. Each generator and fake-score update uses effective batch size 64. Generator and fake-score learning rates are 1.5×10^{-6} and 4×10^{-7}, with AdamW betas $(0, 0.999)$, no weight decay, score-time shift 5, gradient clipping at 10, and generator EMA decay 0.99 starting at iteration 200. The two scores independently load the same 10k-update motion-teacher EMA used by Rolling Forcing. We evaluate the final 1,200-iteration checkpoint. The total training budget includes the preceding 30k supervised updates.

DMD Scoring Window. The real-score and fake-score networks evaluate at most the final 35 latents of each generated sequence. Cropped sequences undergo causal-VAE boundary re-encoding, and shorter sequences are processed with their actual valid lengths.

B External-Method Training and Inference Protocols

All comparison models use HumanML3D+BABEL training data and share frozen evaluation prompts, sequence duration, output frame rate, reference bank, and evaluator weights. Metrics are computed independently within each repeat rather than pooling embeddings. We adapt the external systems to 263-dimensional motion using released code and method-specific rollout objectives. The non-DMD and DMD groups in Table 1 distinguish training objectives; teacher, initialization, rollout windows, effective batch sizes, and update budgets remain method-specific.

TR-off continues post-training without triangular replay under the same backbone and 30k update budget as supervised TR. Gaussian history noise adds 0.05ϵ independently to conditioning history while leaving the supervised band unchanged, with matched initialization, training data, and updates. It serves as a baseline comparison rather than a TR ablation.

Table B1 records the motion-adaptation budgets and evaluated checkpoints. The video-derived students use a motion DiT and motion teacher, not the original video-scale models. The shared bidirectional teacher receives 10k updates; causal AR and ODE initialization stages each use 10k updates where applicable, with 4,096 ODE records. Self Forcing and Rolling Forcing start from the non-causal teacher's ODE distillation; Self Gradient Forcing starts from the causal AR model. For Causal Forcing, we follow the authors' released long-video extension: causal ODE initialization followed by Rolling Forcing training and inference. This is our chosen long-horizon baseline protocol, not a requirement of Causal Forcing itself. Its rolling stage matches our Rolling Forcing adaptation's five-step, five-latent configuration to compare initialization choices. These are fixed adaptation budgets, not claims of convergence at the original video scale.

An iteration denotes the outer DMD iteration, not one generator update. Self Forcing, Rolling Forcing, and the Causal Forcing adaptation use effective batch size 64; Self Gradient Forcing uses 8. Their blocks contain five motion latents (one second at 20 fps), rather than the video implementations' three-latent blocks. NFE (number of function evaluations) counts denoising-network evaluations during inference. We report NFE per committed autoregressive unit: a motion primitive, a single latent token, or a multi-latent chunk, depending on the method. For example, "4 per 5-latent unit" means four denoising evaluations per generated chunk of five latents. MotionStreamer and Resampling Forcing use their predeclared final endpoints of 100k and 31.5k, respectively, without test-based checkpoint selection. The RF endpoint includes its full training curriculum, including the 15-second and history-routing stages.

Table B1: Training and inference protocols for the external 263-dimensional motion adaptations. "AR", "ODE", and "DMD" denote the causal autoregressive-teacher, ODE-initialization, and distribution-matching stages, respectively. NFE counts denoising-network evaluations per committed unit, with each method's unit stated explicitly.

Method	Training Endpoint	Inference NFE per Committed Unit
DART (Zhao et al., 2025)	VAE 200k; denoiser 300k	10 per 8-frame primitive
MotionStreamer (Xiao et al., 2025)	TAE budget 2M; generator 100k	50 per latent token
Resampling Forcing (Guo et al., 2025)	Full curriculum 31.5k	32 per 3-latent unit
Self Forcing (Huang et al., 2025)	ODE 10k; DMD 1k iterations	4 per 5-latent unit
Self Gradient Forcing (Zhuang et al., 2026)	AR 10k; DMD 1.2k iterations	4 per 5-latent unit
Rolling Forcing (Liu et al., 2026)	ODE 10k; DMD 1.2k iterations	5 per 5-latent unit
Causal Forcing Initialization (Zhu et al., 2026)		
+ Rolling Forcing	AR 10k; ODE 10k; rolling DMD 1.2k	5 per 5-latent unit

C Pairwise Preference Evaluation Protocol

Methods and Cases. We compare TR, TR-off, and Rolling Forcing, adapting the pairwise designs of FloodDiffusion (Cai et al., 2026) and DART (Zhao et al., 2025). TR-off is the matched FloodDiffusion baseline post-trained for 30k updates without triangular replay, rather than the unadapted official checkpoint. TR uses the 30k $(\gamma, s) = (0.25, 0.6)$ checkpoint, and Rolling Forcing uses the checkpoint evaluated in Table 1. We select 30 fixed-text prompts from the frozen 256-prompt HumanML3D evaluation manifest by reviewing the text for suitability for sustained motion. We retain ongoing or cyclic actions and exclude one-shot transitions, explicit stopping events, fixed-count or fixed-duration actions, and finite multi-event scripts. The set covers locomotion (10), running (5), dance (4), repetitive exercise (7), and gestures or ground motion (4). Selection uses the text only, not method outputs or metric scores. Table C1 lists the prompts verbatim. Each method supplies one 120-second motion per prompt, reusing available motions from the quantitative evaluation or earlier review renders of the same checkpoints.

Rendering and Presentation. All 90 videos use the same 22-joint skeleton renderer, 960-by-960 resolution, 20 fps, and FloodDiffusion's default 263D skeleton colors. Joint positions are recovered directly from the 263-dimensional motion features, without inverse kinematics, smoothing, or temporal resampling. A fixed camera frames each complete motion above a checkerboard floor; a red trail retains the entire ground-plane root trajectory up to the current frame. Videos contain the prompt and playback time but no method labels. Each pair is concatenated horizontally into one synchronized 120-second video, with the left and right halves identified as A and B. This presentation exposes global trajectory behavior. Figure C1 illustrates the presentation with a frame at 60 seconds for prompt `000467`, selected for the visibility of both figures; annotators view the complete synchronized videos.

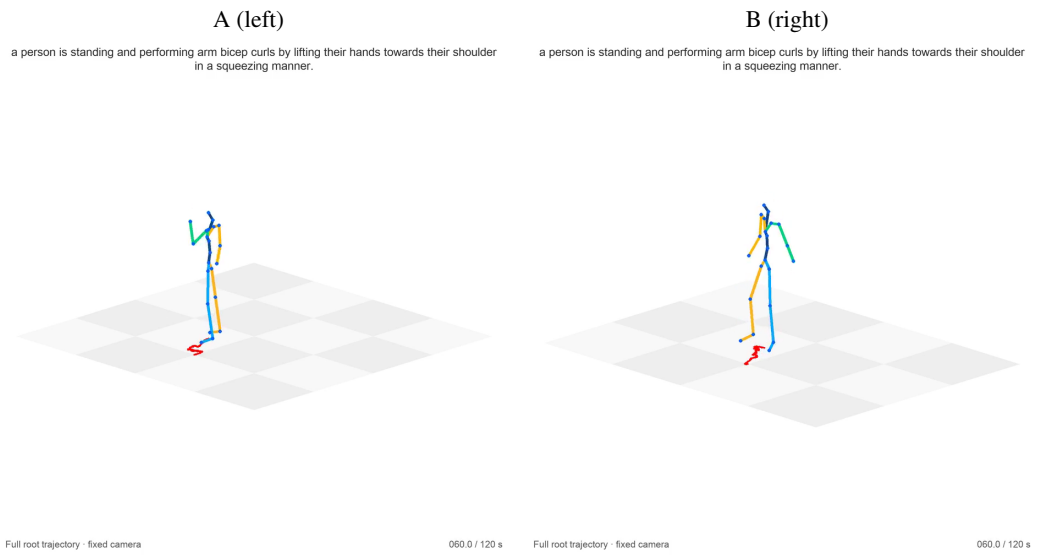


Figure C1: Example comparison video shown to annotators, captured at 60 seconds. A and B identify the left and right videos; method identities are hidden. The frame preserves the original camera views, skeleton colors, and root trajectories.

Pairing and Comparisons. The 90 videos yield 90 within-prompt pairs: TR versus TR-off, TR versus Rolling Forcing, and TR-off versus Rolling Forcing for each prompt. The reported results use all three completed rounds, yielding three judgments per pair and prompt (270 comparisons). Left–right order is pseudorandomized per pair and prompt in the first round, reversed in the second, and restored in the third; method identities remain hidden. All rounds assess the same videos.

Questions and Collection Rules. For each pair, annotators make separate forced A/B choices for overall motion quality and text alignment, yielding 540 binary responses across the 270 comparisons.

Quality concerns natural, coherent movement without conspicuous jitter, foot sliding, freezing, or implausible poses; alignment concerns sustained adherence to the displayed instruction. There is no numerical rating or tie option. Annotators assess the complete video pair while method identities remain hidden.

Analysis. We fit a Bradley–Terry model separately for motion quality and text alignment, with $P(i \succ j) = \exp(b_i)/(\exp(b_i)+\exp(b_j))$. Unregularized maximum likelihood pools all 270 pairwise judgments per criterion across the three rounds (90 per method pair); we report the fitted log-strengths b_i, centered so that $\sum_i b_i = 0$. We fit once to the pooled votes rather than averaging round-wise scores. Higher scores indicate stronger preference in the joint fit, not pairwise win percentages.

Table C1: The 30 prompts selected from the quantitative-evaluation manifest, reproduced verbatim.

No.	Prompt ID	Text
1	000467	a person is standing and performing arm bicep curls by lifting their hands towards their shoulder in a squeezing manner.
2	000556	a person appears to be doing a dance.
3	000710	a person walks forward while twisting their torso side to side.
4	001215	a man sways side by side with arms out
5	001313	a man walks forward while swaying his feet in a zig-zag path.
6	002848	a man is pacing back and forth in a straight line.
7	003005	he does a salsa dance
8	003020	a man slowly sways from side to side, sightly bending his knees.
9	004822	a person is walking in place at a slow pace.
10	005609	a person jogs in place slowly in a counter clockwise circle.
11	006251	the person is walking normally.
12	006523	a person runs forward in a non-linear way.
13	007354	he starts to crawl a lot
14	007418	a person raises the arm and waves multiple times.
15	008100	a person walks in a counter clockwise circle.
16	008296	the man dances his feet in circles in front of himself.
17	009478	a person walks backwards in a straight line
18	009561	a person runs fast diagonal.
19	010114	person is making stepping motion in place
20	011397	a man walks clockwise in a circle
21	012337	a person walks forward carefully placing one foot directly in front of the other foot.
22	012622	a person walking in a diagonal line.
23	013546	this person is seated as if playing the drums.
24	013735	a person giving a round of applause
25	013855	person holds left arm out and right arm forward, then shuffles side to side as if doing the chacha dance
26	014326	person lunges forward with left foot first repeatedly
27	014541	a figure does jumping jacks
28	M006022	someone runs backwards in a clockwise motion.
29	M013023	person moves in a clockwise direction in a circle by sprinting
30	M013619	a man swings his left arm back repeatedly.