DRUM: REAL-TIME DANCING ROBOTS USING MUSIC AND EXECUTION FEEDBACK

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive robot dancing requires a humanoid to respond to musical input throughout a performance while keeping its motion coherent and stable. Two common deployment pipelines fall short of this goal: choreography-specific policies need additional training for new routines, and offline music-to-motion generators can produce references that a general tracker executes poorly. Because both pipelines execute a sequence fixed before playback, upcoming movements are not regenerated when the music changes or the robot deviates from the reference. We present DRUM, a framework for real-time humanoid dancing that combines streaming music-conditioned generation with execution-feedback training. The streaming generator produces successive short dance segments, each conditioned on the incoming music and the motion the robot has just executed. To make these segments compatible with an existing tracker, we post-train the generator with the tracker frozen, alternating reward-based optimization on tracker rollouts with rehearsal on the tracker's executed motion histories. The rewards target what the tracker can execute, and the executed histories target continuation from the states the robot actually reaches. On FineDance and AIST++, the streaming generator before feedback training has a lower kinetic FID and a higher beat alignment score than nine baselines and generates 680 frames per second. On a real X2 humanoid with three frozen trackers, execution-feedback training raises success rates from 0.79–0.93 to 0.91–0.99 at the cost of a higher FID. Real-robot demonstrations show continuous dancing and responsive transitions when users change the music.

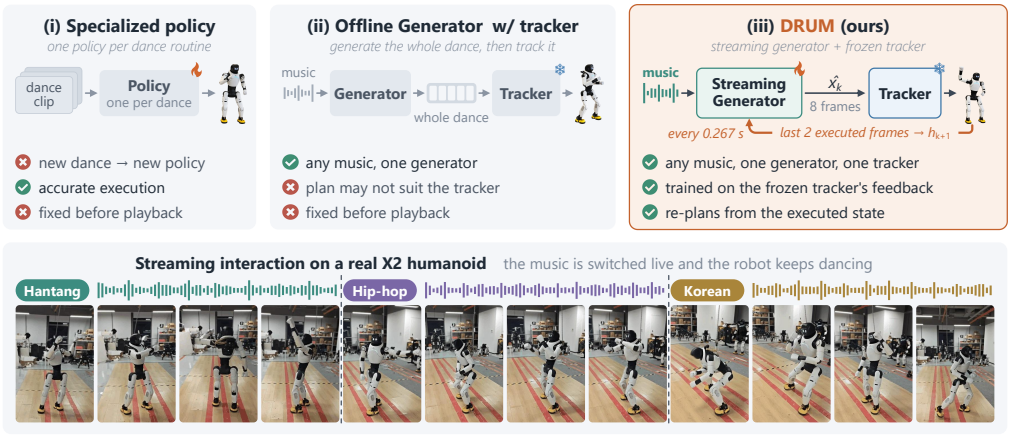


Figure 1: **Robot dancing pipelines.** (i) One specialized policy per routine. (ii) An offline generator plans the whole dance for a tracker. (iii) DRUM streams eight-frame segments conditioned on the music and the last two executed frames. Bottom: a real X2 humanoid keeps dancing as the music is switched live.

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.

1 INTRODUCTION

Music offers an intuitive way to interact with a dancing humanoid (Li et al., 2026b; 2022; Boukheddimi et al., 2022): people can shape its performance by choosing or changing what it hears (Ji et al., 2026), without specifying joint trajectories. Realizing this interaction requires the robot to continuously generate movements (Alexanderson et al., 2023) that follow the music, connect coherently to ongoing motion, and remain executable by its controller (Luo et al., 2026; Li et al., 2026a; RoboParty Lab Team, 2026; Chen et al., 2025; He et al., 2024). Streaming motion generation (Cheng & Huang, 2025; Zhao et al., 2025; Xiao et al., 2025) provides a foundation for responding to changing inputs and recent history. For a physical robot, the central challenge is to keep this generation grounded in what the robot actually executes throughout the performance.

Two representative deployment pipelines illustrate the difficulty. One retargets (Ze et al., 2025b;a) a captured or reconstructed dance to the robot and trains a specialized policy (Liao et al., 2026; Peng et al., 2018) to reproduce it. This can deliver accurate execution, but expanding the repertoire typically requires additional policy training. Another uses an offline music-to-motion generator (Tseng et al., 2023; Li et al., 2023; 2024; 2025b; Yang et al., 2025b) to provide references for a pretrained general-purpose tracker (Chen et al., 2025; Luo et al., 2026). This offers diverse choreography, but the references may be poorly matched to the controller, producing jitter, stumbling, or interrupted motion. Because both pipelines commit to a sequence before playback, upcoming movements also stay unchanged when the user changes the music. Interactive dancing therefore requires references that keep adapting to the music while remaining executable by the same controller.

The mismatch between generation and execution extends beyond individual reference quality. A musically appropriate movement may leave the robot in a different state from the one assumed by the next movement; for example, an incomplete turn changes the starting orientation of the following step. Continuing from the intended trajectory then asks the tracker to absorb both the existing deviation and new motion demands, so local errors can propagate across segments. The learning problem is therefore twofold: generate references suited to the tracker, and generate suitable continuations from the states it actually reaches. The distinction matters over long performances, where each transition inherits the outcome of earlier execution.

Streaming generation (Cai et al., 2026; Tevet et al., 2025) provides the interface for addressing both requirements at inference time. By generating a short segment, executing it, and returning the realized motion as history, the system can update future references with execution feedback. Music specifies how the performance should evolve; history specifies where the next movement begins. To use this interface, the generator needs to learn which references the tracker can realize and how to continue from histories that contain execution deviations. These requirements call for complementary supervision: execution rewards indicate whether a proposed movement suits the controller, and realized histories expose the starting conditions encountered after tracking.

We present **DRUM** (Figure 1), a framework that combines streaming music-conditioned generation with execution-feedback training (EFT). We first train the generator with histories that progressively shift from ground-truth to generated motion. We then freeze a pretrained tracker and alternate reward-based generator adaptation with history-conditioned rehearsal. Execution rewards favor references that the tracker executes well, and rehearsal batches with equal numbers of generated and executed histories train continuation from realized states, with distillation toward the pretrained generator. RLPF (Yue et al., 2025) adapts a motion generator to a frozen tracker with physical-feedback rewards; DRUM couples such adaptation with history-conditioned continuation during streaming musical interaction.

DRUM treats reference compatibility and execution-conditioned continuation as coupled properties learned by the generator. Our contributions are:

- **Streaming music-to-dance generation.** A history-conditioned robot dance model that supports continuous musical interaction and incorporates execution feedback into future generation.
- **Tracker-based execution-feedback training.** Generator post-training that combines execution-reward adaptation with balanced generated-history and executed-history rehearsal while keeping the tracker frozen.

- **Evaluation on a real humanoid.** Against nine baselines on FineDance and AIST++, the streaming generator before feedback training achieves the lowest FID_k and the highest BAS. With three frozen trackers on a real X2 humanoid, execution-feedback training improves success rates and tracking errors, and the robot dances continuously as users change the music.

2 Related Work

2.1 Robot Dancing

Robot dancing has progressed from reproducing prescribed choreography toward more flexible music-driven performance. ASAP (He et al., 2025), BeyondMimic (Liao et al., 2026), and KungfuBot (Xie et al., 2025) demonstrate agile motion through reference tracking, simulation-to-reality alignment, and policy learning, with the movement sequence supplied as a reference. Music-conditioned generation broadens the repertoire: EDGE (Tseng et al., 2023) and Lodge (Li et al., 2024) synthesize human dances that can be converted into robot references, and general trackers such as GMT (Chen et al., 2025) support their execution.

Recent systems bring music closer to robot control. SONIC (Luo et al., 2026) demonstrates music-conditioned dancing with GENMO (Li et al., 2025a), and RoboPerform (Li et al., 2026b) learns audio-conditioned control through teacher-policy distillation. DRUM targets an interactive setting in which users supply music during a performance and a physical humanoid continuously generates and executes short dance segments as the music arrives.

2.2 Streaming Motion Generation

Streaming generation provides a temporal structure for responding to changing inputs. MotionStreamer (Xiao et al., 2025) and A-MDM (Shi et al., 2023) use autoregressive generation for responsive motion synthesis. For audio-driven gestures, Accelerated Rolling Diffusion (Vu et al., 2026) uses progressive denoising, and StreamTalk (Zhang et al., 2026) uses retrieved key-pose anchors to reduce inter-clip drift. DiscoForcing (Ji et al., 2026) streams audio-driven dance with diffusion forcing and deploys it on a humanoid through retargeting and whole-body control. These approaches organize continuity around the synthesized motion. When a robot tracker executes that motion, each segment also produces a physical outcome that can differ from the reference; DRUM conditions the next segment on this realized motion.

2.3 Physical Feedback for Motion Generation

Physical feedback can guide both the choice of a reference and its continuation. PhysDiff (Yuan et al., 2023) incorporates physics-based projection into denoising, and ReinDiffuse (Han et al., 2025) uses physical-plausibility rewards without simulation. RLPF (Yue et al., 2025) adapts a generator using rewards from a frozen tracker, favoring references that the controller executes well. GenTrack (Ling et al., 2026) aligns a text-to-motion generator with group-relative execution rewards, anchoring, and rehearsal, and alternates this alignment with retraining the tracker on generated references. PhyGile (Bao et al., 2026) injects tracker-executed prefixes into text-driven generation, and CLoSD (Tevet et al., 2025) returns executed motion prefixes to a diffusion planner, so future motion depends on actual outcomes. For continuous robot dancing, these roles are coupled: the generator determines what the tracker executes, and execution determines the history for the next prediction. DRUM reflects this coupling in post-training by alternating execution-reward optimization with rehearsal on generated and executed histories, with the tracker fixed throughout.

3 Method

Figure 2 summarizes the framework. A music-conditioned generator proposes a motion segment, a frozen tracker executes it, and the realized motion supplies the next history. We first train a motion variational autoencoder (VAE), then learn a music-conditioned latent diffusion model (LDM) for streaming generation while progressively replacing ground-truth histories with generated histories. We then adapt the generator through execution-reward optimization and history-conditioned

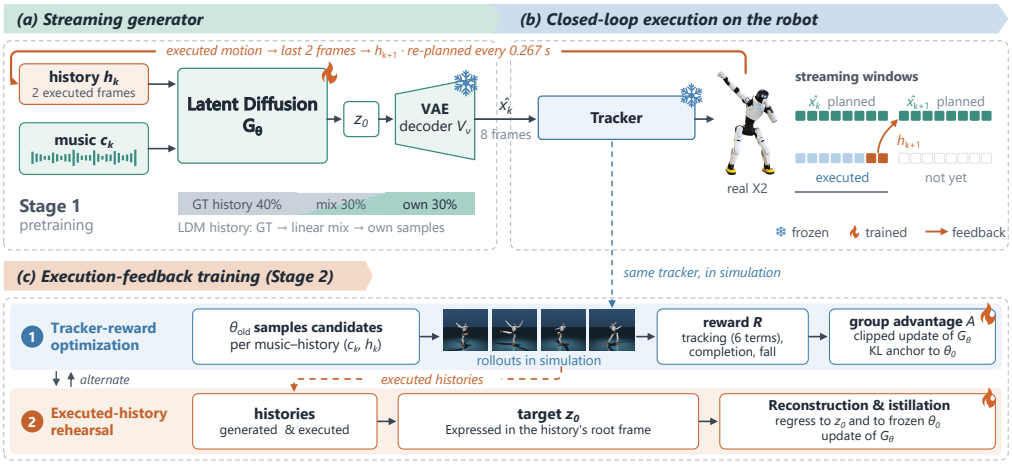


Figure 2: **Overview of DRUM.** (a) A latent diffusion generator predicts an eight-frame segment $\hat{x}_k$ from music c_k and history h_k. (b) A frozen tracker executes $\hat{x}_k$ on the robot, and the last two executed frames form h_{k+1}. (c) Execution-feedback training alternates tracker-reward optimization on simulated rollouts with rehearsal on generated and executed histories; the VAE and tracker stay frozen.

rehearsal. Reward optimization changes the probability of sampled denoising decisions according to tracking performance, while rehearsal distills toward the pretrained generator and trains continuation under execution deviations.

3.1 Streaming Music-Conditioned Dance Generation

Our streaming generator combines a motion VAE Chen et al. (2023) and a music-conditioned latent diffusion model (LDM) Rombach et al. (2022). The VAE encodes an eight-frame future x_0 together with a two-frame history h_k into a 256-dimensional latent; it is pretrained with reconstruction, KL regularization, and kinematic consistency, then frozen. The LDM learns future latents conditioned on music c_k and history:

$$z_0 \sim p_\theta(z_0 \mid a_k), \qquad \hat{x}_k = \mathcal{V}_\nu(s_z z_0, h_k), \qquad a_k = (c_k, h_k). \tag{1}$$

Here, θ denotes LDM parameters, $\mathcal{V}_\nu$ is the fixed VAE decoder, and s_z is the pretrained latent standard deviation used to reverse latent scaling. Each window executes eight new frames at 30 Hz and retains the latest two realized frames, allowing execution feedback to shape the next segment every 0.267 s.

Within the LDM, semi-implicit matching (Cheng & Huang, 2025) supports a short sampling chain while learning the distribution of multiple plausible continuations. The scaled target latent is noised and reconstructed through

$$\begin{aligned} z_t &= \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \\ \hat{z}_0 &= G_\theta(z_t, t, a_k), \qquad \hat{z}_{t-1} \sim q(z_{t-1} \mid z_t, \hat{z}_0) = \mathcal{N}(\mu_\theta, \tilde{\beta}_t I). \end{aligned} \tag{2}$$

The diffusion index is $t = 1, \ldots, T$, β_t is forward variance, and $\bar{\alpha}_t = \prod_{j=1}^{t}(1-\beta_j)$. The predicted clean latent defines the posterior mean μ_θ; its variance is $\tilde{\beta}_t = \beta_t(1-\bar{\alpha}_{t-1})/(1-\bar{\alpha}_t)$, with $\bar{\alpha}_0 = 1$. Sampling starts from standard Gaussian noise and uses $T = 10$ updates, ending with a deterministic step. Music and history remain unnoised.

The data reverse posterior can be multimodal; our single-step transition remains Gaussian for fixed (z_t, a_k). Adversarial matching supervises noisy latent marginals conditioned on (t, a_k). For auxiliary forward diffusion (AFD), sample $z_{t-1} \sim q(\cdot \mid z_0)$, then $z_t \sim \mathcal{N}(\sqrt{1-\beta_t} z_{t-1}, \beta_t I)$, and obtain

$\hat{z}_{t-1}$ from this same z_t:

$$\begin{aligned}\mathcal{L}_D &= -\mathbb{E}[\log D_\psi(z_{t-1}, t, a_k) + \log(1 - D_\psi(\mathrm{sg}(\hat{z}_{t-1}), t, a_k))],\\ \mathcal{L}_{\mathrm{adv}} &= \mathbb{E}[\log(1 - D_\psi(\hat{z}_{t-1}, t, a_k))],\\ \mathcal{L}_{\mathrm{AFD}} &= \mathbb{E}\left[\frac{1-\beta_t}{\beta_t}\|\hat{z}_{t-1} - z_{t-1}\|_2^2\right].\end{aligned} \tag{3}$$

Here, D_ψ classifies forward-data versus generated latents, and sg stops gradients. The discriminator minimizes $\mathcal{L}_D$; generator updates use $\mathcal{L}_{\mathrm{adv}}$ and $\mathcal{L}_{\mathrm{AFD}}$ with D_ψ fixed. Expectations cover examples, uniformly sampled diffusion levels, and noise; AFD sums coordinate errors weighted by the per-step signal-to-noise ratio.

The generator additionally uses latent and decoded-motion Huber reconstruction losses, $\mathcal{L}_{\mathrm{latent}}$ and $\mathcal{L}_{\mathrm{motion}}$, with threshold one. Let $\mathcal{L}_\Delta$ sum three Huber consistency penalties comparing stored increments with increments computed from consecutive frames: root translation, body position, and six-dimensional root rotation. Including the final history frame supervises window continuity. The objective is

$$\mathcal{L}_{\mathrm{pre}} = \mathcal{L}_{\mathrm{latent}} + \mathcal{L}_{\mathrm{motion}} + 10^4\mathcal{L}_\Delta + \lambda_{\mathrm{adv}}\mathcal{L}_{\mathrm{adv}} + \lambda_{\mathrm{AFD}}\mathcal{L}_{\mathrm{AFD}}, \tag{4}$$

where the two λ coefficients weight adversarial and AFD supervision. Gradients pass through the fixed decoder to the LDM.

To prepare autoregressive generation, the first 40% of LDM updates use only ground-truth histories; the next 30% linearly increase the probability of generated histories from zero to one; the final 30% use generated histories only (Bengio et al., 2015). Generated histories come from complete ten-step samples, are detached, and are transformed into the local root frame before reuse. Execution histories are introduced in the subsequent feedback stage.

3.2 Execution-Feedback Training with a Frozen Tracker

Execution reward. A frozen tracker executes candidate references in simulation. For candidate i, segment k, and frame n, let $e^2_{m,i,k,n}$ be the body-averaged squared tracking error for quantity m. We compute

$$\begin{aligned}r_{m,i,k} &= \frac{1}{N_{i,k}}\sum_{n=1}^{N_{i,k}} \exp(-e^2_{m,i,k,n}/\sigma^2_m),\\ R_{i,k} &= \sum_{m=1}^{6} w_m r_{m,i,k} - 5 - (1 - c_{i,k}) - 2f_{i,k}.\end{aligned} \tag{5}$$

Here, $N_{i,k}$ counts evaluated frames, and $c_{i,k}$ is the available-to-planned simulation-step ratio for the segment, capped at one. The flag $f_{i,k} = 1$ records simulator termination or a reference-aware fall: when reference pelvis height is at least 0.50 m and its up-axis vertical component at least 0.50, executed height falls over 0.25 m below reference or its up-axis component drops below 0.30. The six terms measure pelvis position/orientation, heading-local body position/orientation, and world-frame body linear/angular velocity over 14 representative bodies. Weights are $(0.5, 0.5, 1, 1, 1, 1)$; scales are $(0.3\,\mathrm{m}, 0.4\,\mathrm{rad}, 0.3\,\mathrm{m}, 0.4\,\mathrm{rad}, 1\,\mathrm{m/s}, 3.14\,\mathrm{rad/s})$. Orientations use angular distance. Pelvis position removes the initial horizontal offset; local body quantities use each trajectory's own pelvis origin and heading. Fewer than two valid frames yields $R = -8$.

Reward-driven distribution adaptation. For each new candidate group, θ_{old} snapshots the current generator and samples four candidates per music–history condition; its recorded log probabilities stay fixed for all four replay updates. Each candidate is rolled out in closed loop for 48 segments (12.8 s): the last two frames that the tracker executes for one segment form the history of the next. Segment advantages are normalized within each group (Shao et al., 2024), $A_{i,k} = (R_{i,k} - \bar{R}_k)/(s_k + 10^{-4})$, where $\bar{R}_k$ and s_k are the within-group reward mean and population standard deviation. For stored stochastic transitions ($t \geq 2$), we recompute the dimension-averaged Gaussian log probability and its normalized ratio:

$$\ell_\theta = \frac{1}{D}\log\mathcal{N}(z_{t-1}; \mu_\theta(z_t, t, a_k), \tilde{\beta}_t I), \qquad \rho_\theta = e^{\ell_\theta - \ell_{\mathrm{old}}}, \quad D = 256. \tag{6}$$

Using the same saved inputs and outputs, fixed variance, and disabled dropout, we optimize a clipped objective (Schulman et al., 2017; Black et al., 2024):

$$\begin{aligned}\mathcal{L}_{\text{reward}} = & -\mathbb{E}[\min(\rho_\theta A_{i,k}, \text{clip}(\rho_\theta, 0.8, 1.2)A_{i,k})] \\ & + 0.02\mathbb{E}[e^\delta - \delta - 1], \qquad \delta = \ell_{\theta_0} - \ell_\theta.\end{aligned} \tag{7}$$

The frozen initial generator θ_0 supplies the KL-type anchor. Expectations average replayed transitions. Gradients flow through the current denoising mean only; rewards, advantages, stored states, and tracker execution are detached. This raises the probability of better-tracked samples while anchoring the generator to its pretrained distribution.

History-conditioned rehearsal. Reward updates alternate with supervised rehearsal. For a noised VAE target z_t, define $g_\theta = G_\theta(z_t, t, a_k)$ and $g_0 = G_{\theta_0}(z_t, t, a_k)$. We retain latent reconstruction and fixed-model distillation:

$$\mathcal{L}_{\text{reh}} = \mathbb{E}\left[\text{mean}[(g_\theta - z_0)^2] + \text{mean}[(g_\theta - \text{sg}(g_0))^2]\right]. \tag{8}$$

Each mean averages squared errors over latent coordinates; sg stops gradients. Each batch contains equal numbers of generated-history and executed-history examples. For rehearsal, the GT future at the same music timestamps is expressed in the selected history's local root frame and re-encoded with that history by the frozen VAE to obtain z_0. Generated histories match the pretraining setting, and executed histories expose execution deviations. At deployment, the latest two executed frames condition the next window. The VAE and tracker remain frozen. Feedback training runs for 18 rounds with 1,080 reward and 1,080 rehearsal updates; Appendix A gives the full configuration.

4 Experiments

4.1 Experimental Setup

Robot Platform. We conduct experiments on the AGIBOT Lingxi X2 humanoid (about 1.4 m tall, 40 kg, 29 controlled joints). The frozen tracker executes the 30 Hz references at 50 Hz after time interpolation.

Datasets. We use FineDance (Li et al., 2023) and AIST++ (Li et al., 2021) for music-conditioned dance generation. The public FineDance subset contains approximately 7.7 hours of motion at 30 FPS (about 831,600 frames) across 16 dance genres (Li et al., 2024). AIST++ contains 1,408 sequences totaling approximately 5.2 hours at 60 FPS, covering 10 genres and 30 dancers. Both datasets are retargeted to X2 and represented at 30 Hz while preserving music–motion synchronization. Following the data split protocol of FlowerDance Yang et al. (2025b), we partition the training, validation, and test sets by music identity before window extraction, ensuring that no music track is shared across splits. Normalization statistics are computed using the training set only.

Evaluation Protocol. All baselines and DRUM are retrained on the same retargeted X2 motion data in a shared robot-motion representation, and each method uses the checkpoint with the best validation score under the same criterion. Retargeting is performed offline, so DRUM generates robot references directly at inference. Generation is evaluated without tracker execution using kinetic and geometric Fréchet distances (FID_k, FID_g), diversity (DIV_k, DIV_g), Beat Alignment Score (BAS) (Li et al., 2021; 2024), and Foot Sliding Rate (FSR). On the robot, each frozen tracker executes all generators from matched initial states, and we report success rate (Succ), joint-angle error E_{joint} (rad), and key-body position error E_{key} (m) over 14 bodies. Appendix B defines the metrics.

4.2 Dance Generation Evaluation

Table 1 compares DRUM with nine music-to-dance generators: DanceNet (Zhuang et al., 2022), FACT (Li et al., 2021), MNET (Kim et al., 2022), Bailando (Siyao et al., 2022), EDGE (Tseng et al., 2023), FineNet (Li et al., 2023), Lodge (Li et al., 2024), MEGADance (Yang et al., 2025a), and FlowerDance (Yang et al., 2025b).

Table 1: **Music-conditioned dance generation.** All methods are retrained on the same retargeted X2 data. w/o EFT: before execution-feedback training; DRUM: after EFT with UFO-rot6d; RT Stream: real-time streaming; FPS excludes the tracker. **Bold**/underline: best/second best excluding ground truth (ties share a mark).

Method	RT Stream	FineDance							AIST++				
		FIDk ↓	FIDg ↓	DIVk ↑	DIVg ↑	BAS ↑	FSR ↓	FPS ↑	FIDk ↓	FIDg ↓	DIVk ↑	DIVg ↑	BAS ↑
Ground Truth	—	—	—	9.17	7.20	0.200	0.194	—	—	—	8.11	7.33	0.224
DanceNet	✗	143.82	119.21	2.87	2.77	0.149	0.402	—	92.72	60.14	2.48	2.30	0.129
FACT	✗	119.16	98.93	3.01	4.92	0.155	0.372	26	75.93	48.91	5.44	6.02	0.214
MNET	✗	97.41	92.10	3.09	5.71	0.162	0.380	28	79.42	55.19	4.98	5.29	0.209
Bailando	✗	92.28	61.44	7.19	6.09	0.196	0.242	194	39.79	18.50	6.89	6.02	0.219
EDGE	✗	78.82	39.82	**7.76**	6.32	0.218	0.194	112	31.66	20.71	3.74	4.69	<u>0.228</u>
FineNet	✗	70.41	26.84	6.04	5.82	0.221	0.181	220	29.62	<u>16.18</u>	5.94	5.28	<u>0.228</u>
Lodge	✗	63.92	32.80	5.72	5.28	0.220	<u>0.099</u>	228	41.22	20.24	5.81	5.22	0.171
MEGADance	✗	55.91	**22.43**	5.20	6.04	0.201	0.252	240	32.81	18.46	5.76	5.49	0.227
FlowerDance	✗	<u>33.92</u>	24.73	6.74	**7.22**	<u>0.226</u>	0.176	**1840**	<u>24.20</u>	16.72	<u>7.03</u>	<u>6.77</u>	0.213
DRUM (Ours) w/o EFT	✓	**28.84**	<u>23.16</u>	<u>7.42</u>	<u>6.86</u>	**0.228**	0.141	<u>680</u>	**18.84**	**15.52**	**7.19**	**6.84**	**0.232**
DRUM (Ours)	✓	39.76	28.81	6.98	6.27	0.225	**0.088**	<u>680</u>	27.19	22.41	6.82	6.09	0.219

Generation quality. Before execution-feedback training, DRUM (w/o EFT) has the lowest FID_k on both datasets (28.84 on FineDance and 18.84 on AIST++), 15.0% and 22.1% below the next best method, FlowerDance (33.92 and 24.20). It ranks first in FID_g on AIST++ (15.52 vs. 16.18 for FineNet) and second on FineDance (23.16 vs. 22.43 for MEGADance). Its diversity ranks first or second on both datasets (DIV_k of 7.42 and 7.19; ground truth 9.17 and 8.11), and it has the highest BAS on both (0.228 and 0.232).

Effect of execution-feedback training. EFT trades distribution similarity for physical plausibility. FID_k rises from 28.84 to 39.76 on FineDance and from 18.84 to 27.19 on AIST++, while FSR on FineDance falls from 0.141 to 0.088, the lowest among all methods. DIV_k decreases from 7.42 to 6.98 on FineDance and from 7.19 to 6.82 on AIST++, and BAS from 0.228 to 0.225 and from 0.232 to 0.219.

Real-time streaming. We call a generator streaming if it consumes music incrementally, outputs successive segments, and keeps its 95th-percentile update time below the duration of the emitted motion. Among the evaluated methods, only DRUM meets these criteria, generating 680 frames per second on FineDance, more than 22 times the 30 Hz playback rate. FlowerDance is faster offline (1,840 FPS) but takes the complete song as input, and FACT (26 FPS) and MNET (28 FPS) are slower than playback.

4.3 Real-World Feedback Evaluation

Setup. We deploy three frozen trackers on X2: SONIC (Luo et al., 2026), BFM-Zero (Li et al., 2026a), and UFO-rot6d (RoboParty Lab Team, 2026), all trained on LAFAN1 (Harvey et al., 2020), SEED, FineDance, EDGE, and AMASS (Mahmood et al., 2019). Each tracker executes references from four generators at 50 Hz under matched music, initial states, and durations: EDGE, FlowerDance, DRUM w/o EFT, and DRUM after 18 rounds of execution-feedback training with that tracker.

Results. Table 2 shows three findings. *Generation quality alone does not guarantee execution quality.* DRUM w/o EFT has the best FID_k and BAS, yet before feedback training it does not lead every execution metric; for example, FlowerDance has a lower E_{joint} with SONIC on FineDance (0.31 vs. 0.33) and with BFM-Zero on AIST++ (0.38 vs. 0.40), and a higher success rate with UFO-rot6d on AIST++ (0.85 vs. 0.83). This motivates feedback from the tracker itself.

EFT improves execution with each tracker. Within each tracker group, DRUM has the highest success rate and the lowest E_{joint} and E_{key} on both datasets. Across trackers and datasets, success rises from 0.79–0.93 to 0.91–0.99, and on FineDance E_{key} decreases by 25.0%, 26.3%, and 38.9% with SONIC, BFM-Zero, and UFO-rot6d.

Table 2: **Real-robot execution on X2.** Four generators per frozen tracker under matched conditions; DRUM is adapted to each tracker. **Bold**/underline: best/second best per tracker (ties share a mark).

Frozen tracker	Generator	FineDance			AIST++		
		Succ ↑	E_{joint} (rad) ↓	E_{key} (m) ↓	Succ ↑	E_{joint} (rad) ↓	E_{key} (m) ↓
SONIC	EDGE	0.77	0.46	0.41	0.64	0.51	0.49
	FlowerDance	0.85	0.31	0.26	0.82	0.44	0.31
	DRUM w/o EFT	0.86	0.33	0.24	0.81	0.41	0.30
	DRUM	**0.98**	**0.27**	**0.18**	**0.95**	**0.29**	**0.22**
BFM-Zero	EDGE	0.74	0.41	0.38	0.66	0.49	0.44
	FlowerDance	0.88	0.30	0.22	0.79	0.38	0.29
	DRUM w/o EFT	0.89	0.30	0.19	0.79	0.40	0.31
	DRUM	**0.97**	**0.20**	**0.14**	**0.91**	**0.26**	**0.21**
UFO-rot6d	EDGE	0.78	0.48	0.34	0.72	0.45	0.38
	FlowerDance	0.89	0.24	0.20	0.85	0.39	0.26
	DRUM w/o EFT	0.93	0.27	0.18	0.83	0.37	0.22
	DRUM	**0.99**	**0.16**	**0.11**	**0.98**	**0.22**	**0.19**

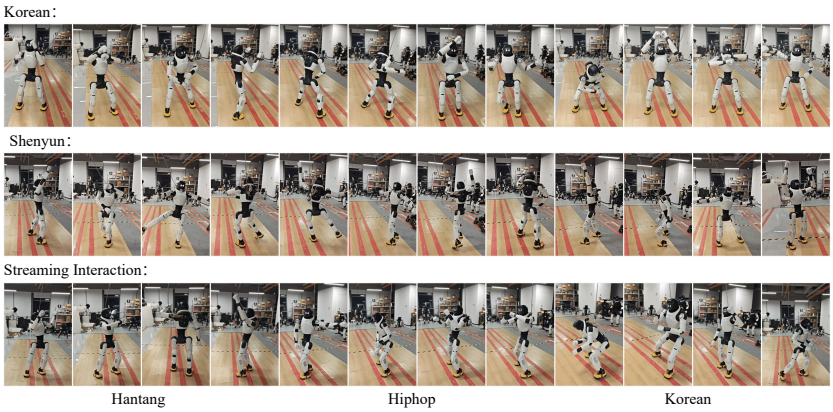


Figure 3: **Real-robot performances** of DRUM with the frozen UFO-rot6d tracker on X2. Top two rows: performances to Korean and Shenyun music. Bottom: the music is switched live from Hantang to hip-hop to Korean.

DRUM with UFO-rot6d gives the best execution. Among the adapted configurations, DRUM with UFO-rot6d leads all six execution metrics. We use this pair for the user study and the real-robot demonstrations in Figure 3, which include live switches between songs; the supplementary videos contain the complete performances.

4.4 Ablation Studies

Streaming generator design (Group I). Table 3 ablates the streaming generator (Section 3.1) and execution-feedback training (Section 3.2) on FineDance. The history schedule has the largest effect on FID_k. Training with ground-truth histories only (I-a) gives the highest FID_k (37.71 vs. 28.84 for I-e) and the lowest BAS (0.204), consistent with exposure bias: the model is not trained on its own outputs, which it conditions on at inference. Using generated histories from the start (I-b) gives an FID_k of 32.19, 11.6% above the full generator. Removing semi-implicit matching (I-c), i.e., $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{AFD}}$, raises FID_k to 30.94 and gives the lowest DIV_k (7.10), consistent with mode averaging under a ten-step sampler. Removing $\mathcal{L}_{\Delta}$ (I-d) gives the highest FSR (0.188 vs. 0.141) and lowers BAS to 0.207, while DIV_k rises to 7.90.

Execution-feedback training (Group II). Reward optimization provides most of the execution gain: reward only (II-b) raises Succ from 0.93 to 0.99 and lowers E_{joint} from 0.27 to 0.10, while

Table 3: **Ablations on FineDance.** Group I: streaming generator. Group II: feedback training, executed on X2 with UFO-rot6d. **Bold**/underline: best/second best per group (ties share a mark).

Group I: Streaming Generator						Group II: Execution-Feedback Training					
	Configuration	FIDk↓	DIVk↑	BAS↑	FSR↓		Configuration	FIDk↓	FSR↓	Succ↑	E_{joint}↓	E_{key}↓
I-a	GT history only	37.71	7.79	0.204	0.176	II-a	w/o EFT (θ_0)	**28.84**	0.141	0.93	0.27	0.18
I-b	Gen. history from start	32.19	7.38	0.220	0.159	II-b	Reward only	40.19	0.096	**0.99**	**0.10**	0.12
I-c	w/o semi-implicit	30.94	7.10	0.226	0.162	II-c	Rehearsal only	35.62	0.160	0.91	0.30	0.17
I-d	w/o $\mathcal{L}_\Delta$	31.85	**7.90**	0.207	0.188	II-d	w/o executed history	33.70	0.118	0.94	0.25	0.20
I-e	Full generator	**28.84**	7.42	**0.228**	**0.141**	II-e	DRUM (full)	39.76	**0.088**	**0.99**	0.16	**0.11**

Table 4: **User study** on X2 with UFO-rot6d (40 participants; 5-point Likert scale; mean ± std). **Bold**/underline: best/second best.

Method	Naturalness ↑	Anthropomorphism ↑	Stability ↑	Music Alignment ↑
EDGE	3.80 ± 0.08	3.22 ± 0.11	3.16 ± 0.06	3.77 ± 0.09
FlowerDance	3.84 ± 0.10	3.29 ± 0.07	3.21 ± 0.11	3.80 ± 0.05
DRUM w/o EFT	4.02 ± 0.08	$\mathbf{3.30 \pm 0.04}$	3.20 ± 0.12	$\mathbf{3.92 \pm 0.03}$
DRUM (Ours)	$\mathbf{4.09 \pm 0.05}$	3.28 ± 0.09	$\mathbf{3.97 \pm 0.06}$	3.89 ± 0.05

FID_k rises from 28.84 to 40.19. **Qualitatively, reward-only training overemphasizes balance, causing the robot to favor near-stationary poses with small-amplitude movements and lose its rich dance repertoire; despite improved execution metrics, visual quality deteriorates markedly, highlighting the importance of rehearsal in preserving dance expressiveness.** Rehearsal alone (II-c) lowers Succ to 0.91, raises E_{joint} to 0.30 and FID_k to 35.62, and lowers E_{key} slightly (0.17 vs. 0.18); its loss omits the adversarial, AFD, and continuity terms of pretraining (Eq. 4). Rehearsing on generated histories only (II-d) keeps FID_k lower (33.70 vs. 39.76) but loses most of the execution gain (Succ 0.94; E_{key} 0.20, above the 0.18 without EFT), whereas including executed histories preserves it (II-e). The full model matches reward only in Succ (0.99) and has the lowest FSR (0.088) and E_{key} (0.11), with a higher E_{joint} (0.16 vs. 0.10).

4.5 User Study

Forty participants with backgrounds in humanoid robotics or digital-human animation rated anonymized real-robot videos of EDGE, FlowerDance, DRUM w/o EFT, and DRUM, all executed on X2 by the frozen UFO-rot6d tracker and shown in random order for each music clip. Ratings used a 5-point Likert scale (1 = worst, 5 = best) for Naturalness (smooth motion without jitter, sudden jumps, or unnatural poses), Anthropomorphism (human-like expressiveness, rhythm, and coordination), Stability (balance without stumbles, falls, or protective stops), and Music Alignment (synchrony with beat, tempo, and style).

Table 4 shows the largest difference in Stability: DRUM scores 3.97 versus 3.16–3.21 without EFT, and its gain over DRUM w/o EFT (3.20) matches the success-rate gain from EFT. DRUM also has the highest Naturalness (4.09 vs. 3.80–4.02). DRUM w/o EFT has the highest Music Alignment (3.92 vs. 3.89 for DRUM), consistent with its higher BAS, and Anthropomorphism is similar across methods (3.22–3.30).

5 Conclusion and Limitations

We presented DRUM, a framework for real-time robot dancing that connects streaming music-conditioned generation with tracker-based execution feedback. By alternating execution-reward optimization with rehearsal on realized motion histories, DRUM adapts the generator to a frozen tracker. On a real X2 humanoid, this improves success rates and tracking errors with three trackers while supporting continuous musical interaction. These gains come with a higher FID_k, as the generated distribution moves toward references that the tracker executes well. Our evaluation covers a single robot platform, and transferring DRUM to another embodiment requires renewed generator adaptation; future work will develop an adaptive framework that supports multiple embodiments.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and translation of the manuscript. We have not used generative AI tools for research ideation, method design, experiments, or data analysis. We have reviewed all AI-assisted text, references, and figures against our experimental records, and we take responsibility for the final content of this work.

ETHICS STATEMENT

We have not identified any specific ethical concerns arising from this work.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide detailed descriptions of the dataset construction procedure and the computation of the evaluation metrics in the paper and accompanying materials. These descriptions document the steps used to prepare the data and evaluate the results, supporting the replication of our experimental evaluation.

REFERENCES

Simon Alexanderson, Rajmund Nagy, Jonas Beskow, and Gustav Eje Henter. Listen, denoise, action! audio-driven motion synthesis with diffusion models. *ACM Transactions on Graphics (TOG)*, 42 (4):1–20, 2023.

Jiacheng Bao, Haoran Yang, Yucheng Xin, Junhong Liu, Yuecheng Xu, Han Liang, Pengfei Han, Xiaoguang Ma, Dong Wang, and Bin Zhao. PhyGile: Physics-Prefix Guided Motion Generation for Agile General Humanoid Motion Tracking, 2026. URL https://arxiv.org/abs/2603.19305.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In *Advances in Neural Information Processing Systems*, 2015.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training Diffusion Models with Reinforcement Learning. In *International Conference on Learning Representations*, 2024.

Melya Boukheddimi, Daniel Harnack, Shivesh Kumar, Rohit Kumar, Shubham Vyas, Octavio Arriaga, and Frank Kirchner. Robot dance generation with music based trajectory optimization. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3069–3076, 2022.

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.

Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18000–18010. IEEE, 2023.

Zixuan Chen, Mazeyu Ji, Xuxin Cheng, Xuanbin Peng, Xue Bin Peng, and Xiaolong Wang. GMT: General Motion Tracking for Humanoid Whole-Body Control, 2025. URL https://arxiv.org/abs/2506.14770.

Yongkang Cheng and Shaoli Huang. HoloGest: Decoupled diffusion and motion priors for generating holisticly expressive co-speech gestures. In *2025 International Conference on 3D Vision (3DV)*, pp. 748–757, 2025.

Gaoge Han, Mingjiang Liang, Jinglei Tang, Yongkang Cheng, Wei Liu, and Shaoli Huang. ReinDiffuse: Crafting Physically Plausible Motions with Reinforced Diffusion Model. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2025. URL https://arxiv.org/abs/2410.07296.

Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. Robust Motion Inbetweening. *ACM Transactions on Graphics*, 2020.

Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. OmniH2O: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, Zeji Yi, Guannan Qu, Kris Kitani, Jessica Hodgins, Linxi Fan, Yuke Zhu, Changliu Liu, and Guanya Shi. ASAP: Aligning Simulation and Real-World Physics for Learning Agile Humanoid Whole-Body Skills. In *Robotics: Science and Systems*, 2025. URL https://arxiv.org/abs/2502.01143.

Kaiyang Ji, Bingsheng Qian, Binghuan Wu, Kangyi Chen, Ye Shi, and Jingya Wang. DiscoForcing: A Unified Framework for Real-Time Audio-Driven Character Control with Diffusion Forcing, 2026. URL https://arxiv.org/abs/2605.28491.

Jinwoo Kim, Heeseok Oh, Seongjean Kim, Hoseok Tong, and Sanghoon Lee. A Brand New Dance Partner: Music-Conditioned Pluralistic Dancing Controlled by Multiple Dance Genres. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Jiefeng Li, Jinkun Cao, Haotian Zhang, Davis Rempe, Jan Kautz, Umar Iqbal, and Ye Yuan. GENMO: A GENeralist Model for Human MOtion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025a.

Ronghui Li, Junfan Zhao, Yachao Zhang, Mingyang Su, Zeping Ren, Han Zhang, Yansong Tang, and Xiu Li. FineDance: A fine-grained choreography dataset for 3D full body dance generation. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10200–10209, 2023.

Ronghui Li, YuXiang Zhang, Yachao Zhang, Hongwen Zhang, Jie Guo, Yan Zhang, Yebin Liu, and Xiu Li. Lodge: A Coarse to Fine Diffusion Network for Long Dance Generation Guided by the Characteristic Dance Primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. URL https://arxiv.org/abs/2403.10518.

Ronghui Li, Hongwen Zhang, Yachao Zhang, Yuxiang Zhang, Youliang Zhang, Jie Guo, Yan Zhang, Xiu Li, and Yebin Liu. Lodge++: High-quality and long dance generation with robust choreography patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. AI choreographer: Music conditioned 3D dance generation with AIST++. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13381–13392, 2021.

Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, et al. BFM-Zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 79697–79725, 2026a.

Yizhu Li, Nan Liang, Meysam Effati, and Goldie Nejat. Dances with social robots: A pilot study at long-term care. *Robotics*, 11(5):96, 2022.

Zhe Li, Cheng Chi, Yangyang Wei, Boan Zhu, Tao Huang, Zhenguo Sun, Yibo Peng, Pengwei Wang, Zhongyuan Wang, Fangzhou Liu, Chang Xu, and Shanghang Zhang. Do you have freestyle? Expressive humanoid locomotion via audio control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 956–965, June 2026b.

Qiayuan Liao, Takara E Truong, Xiaoyu Huang, Yuman Gao, Guy Tevet, Koushil Sreenath, and C Karen Liu. BeyondMimic: From motion tracking to versatile humanoid control via guided diffusion. *Science Robotics*, 11(117):eadx8924, 2026.

Zeyu Ling, Xinyao Yu, Renye Yan, Jikang Cheng, Zhanke Wang, Qing Shuai, and Changqing Zou. GenTrack: Physical Alignment for Robot-Native Motion Generation and Zero-Shot Humanoid Tracking, 2026. URL https://arxiv.org/abs/2608.01410.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, et al. SONIC: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11(117):eaed4592, 2026.

Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of Motion Capture as Surface Shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions On Graphics (TOG)*, 37(4):1–14, 2018.

RoboParty Lab Team. UFO: An unsupervised reinforcement learning framework for humanoid control. https://github.com/Roboparty/UFO, 2026. Project page: https://roboparty.github.io/UFO/.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10674–10685. ieee, 2022.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024.

Yi Shi, Jingbo Wang, Xuekun Jiang, Bingkun Lin, Bo Dai, and Xue Bin Peng. Interactive Character Control with Auto-Regressive Motion Diffusion Models, 2023. URL https://arxiv.org/abs/2306.00416.

Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando: 3D Dance Generation by Actor-Critic GPT with Choreographic Memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Guy Tevet, Sigal Raab, Setareh Cohan, Daniele Reda, Zhengyi Luo, Xue Bin Peng, Amit Haim Bermano, and Michiel van de Panne. CLoSD: Closing the Loop between Simulation and Diffusion for Multi-task Character Control. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZISppZSTv.

Jonathan Tseng, Rodrigo Castellon, and C. Karen Liu. EDGE: Editable Dance Generation from Music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. URL https://edge-dance.github.io/.

Evgeniia Vu, Andrei Boiarov, and Dmitry Vetrov. Streaming Generation of Co-Speech Gestures via Accelerated Rolling Diffusion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/39807.

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 (ICCV)*, pp. 10086–10096, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Xiao_MotionStreamer_Streaming_Motion_Generation_via_Diffusion-based_Autoregressive_Model_in_Causal_ICCV_2025_paper.html.

Weiji Xie, Jinrui Han, Jiakun Zheng, Huanyu Li, Xinzhe Liu, Jiyuan Shi, Weinan Zhang, Chenjia Bai, and Xuelong Li. KungfuBot: Physics-Based Humanoid Whole-Body Control for Learning Highly-Dynamic Skills. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.12851.

Kaixing Yang, Xulong Tang, Ziqiao Peng, Yuxuan Hu, Jun He, and Hongyan Liu. MEGADance: Mixture-of-Experts Architecture for Genre-Aware 3D Dance Generation. In *Advances in Neural Information Processing Systems*, 2025a.

Kaixing Yang, Xulong Tang, Ziqiao Peng, Xiangyue Zhang, Puwei Wang, Jun He, and Hongyan Liu. FlowerDance: MeanFlow for efficient and refined 3D dance generation. *arXiv preprint arXiv:2511.21029*, 2025b.

Ye Yuan, Jiaming Song, Umar Iqbal, Arash Vahdat, and Jan Kautz. PhysDiff: Physics-Guided Human Motion Diffusion Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://nvlabs.github.io/PhysDiff/.

Junpeng Yue, Zepeng Wang, Yuxuan Wang, Weishuai Zeng, Jiangxing Wang, Xinrun Xu, Yu Zhang, Sipeng Zheng, Ziluo Ding, and Zongqing Lu. RL from Physical Feedback: Aligning Large Motion Models with Humanoid Control, 2025. URL https://arxiv.org/abs/2506.12769.

Yanjie Ze, Zixuan Chen, João Pedro Araújo, Zi-ang Cao, Xue Bin Peng, Jiajun Wu, and C Karen Liu. TWIST: Teleoperated whole-body imitation system. *arXiv preprint arXiv:2505.02833*, 2025a.

Yanjie Ze, Siheng Zhao, Weizhuo Wang, Angjoo Kanazawa, Rocky Duan, Pieter Abbeel, Guanya Shi, Jiajun Wu, and C Karen Liu. TWIST2: Scalable, portable, and holistic humanoid data collection system. *arXiv preprint arXiv:2511.02832*, 2025b.

Xiangyue Zhang, Jianfang Li, Jiaxu Zhang, Kaixing Yang, and Steven Hoi. StreamTalk: Streaming Co-Speech Gesture Generation with Key-Pose Anchoring. In *European Conference on Computer Vision*, 2026. URL https://xiangyuezhang.com/StreamTalk/.

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*, volume 2025, pp. 23569–23592, 2025.

Wenlin Zhuang, Congyi Wang, Jinxiang Chai, Yangang Wang, Ming Shao, and Siyu Xia. Music2Dance: DanceNet for Music-Driven Dance Generation. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2022.

A Training Details

Generative model configuration and training. We use 30 Hz motion with 372 features, 438-dimensional music features, two history frames, and eight generated frames per window. The VAE has seven layers, width 256, and one 256-dimensional latent token; the LDM has eight layers and width 512. Both use feed-forward width 1024, four attention heads, and dropout 0.1. LDM sampling uses ten steps with the cosine noise schedule (offset 0.008, maximum variance 0.999). Bayesian selection within semi-implicit training gives approximately one discriminator update per five generator updates. VAE and LDM training use 400,000 and 600,000 optimizer updates, with global batches 128 and 8,192 on eight GPUs. VAE history stages span 200,000/100,000/100,000 updates; the LDM's ground-truth, transition, and generated-history stages span 240,000/180,000/180,000 updates. Both use AdamW with learning rate 10^{-4}, betas $(0.9, 0.999)$, weight decay 0.01, linear learning-rate decay, gradient clipping at 1.0, and EMA 0.999. Each step updates one window batch; random crop sampling is budgeted in optimizer updates rather than epochs. The VAE remains frozen after pretraining.

Feedback training parameters. Feedback training runs for 18 rounds. Each round collects 15 groups, each containing two history-conditioned cases and four candidates per case. Each candidate comprises 48 eight-frame segments (12.8 s). Four reward updates per group yield 60 reward updates per round, interleaved with 60 supervised rehearsal updates. Across 18 rounds, this gives 1,080 reward updates and 1,080 rehearsal updates, totaling 2,160 generator optimizer steps. Each reward update replays eight segments per candidate and one stochastic diffusion transition per segment, giving 64 transitions; rehearsal batches contain two examples. Every batch contains equal numbers of generated-history and executed-history examples. AdamW uses learning rate 5×10^{-6}, betas $(0.9, 0.99)$, zero weight decay, and gradient clipping at 1.0; policy clipping, anchor weight, advantage stabilizer, and distillation weight are 0.2, 0.02, 10^{-4}, and 1.0, respectively. The VAE, tracker, and diffusion variances remain fixed. The tracker runs at 50 Hz with time-interpolated 30 Hz references.

B EVALUATION DETAILS

Generation metrics are computed on generated motion without tracker execution, using identical test music, durations, and feature extraction for all methods; test data are not used for model selection. DIV_k and DIV_g are average pairwise feature distances, and FSR is the fraction of frames in which a contact foot slides faster than a velocity threshold. Throughput is measured on the same hardware with batch size one, excluding the tracker. E_{joint} is the wrapped joint-angle error. E_{key} is aligned only at the starting horizontal position, so accumulated drift is retained. Both errors follow the MPJPE and MPKPE definitions of RoboPerform (Li et al., 2026b), include valid frames from unsuccessful trials, and are averaged over sequences and repetitions for each dataset.