AHEADance: Anticipatory Humanoid Dance from Streaming Audio

Tianhu Peng*, Yukun Wang*, Lingfan Bao, Yiyang Jia and Chengxu Zhou†

https://ihcr.github.io/AHEADance

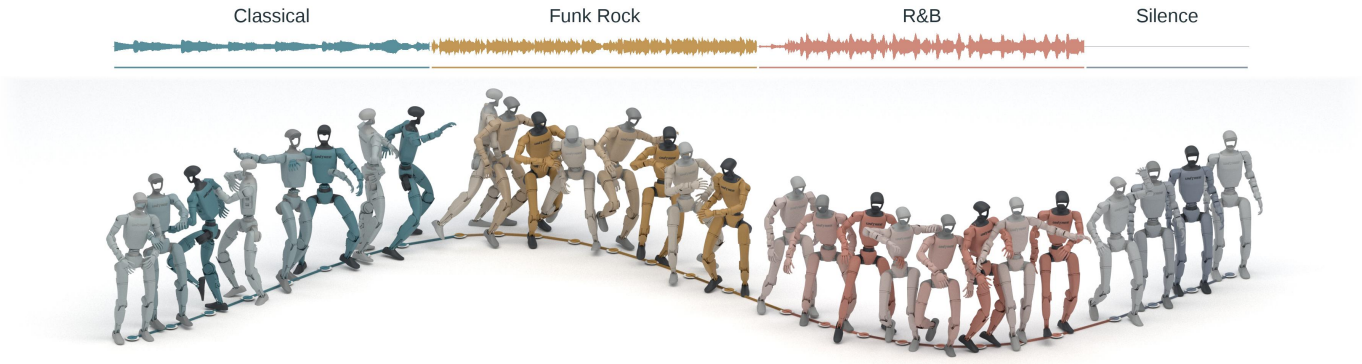


Fig. 1. **Music-driven dance across styles and silence.**

***Abstract*—A humanoid that dances to streamed music must commit each motion chunk before it has heard the music that chunk accompanies, and must continue from motion it has already committed. We present AHEADance, a robot-native streaming dance generator executed on a humanoid robot by a whole-body tracker. ListenAhead conditions each chunk on heard music and a frozen music model's forecasts of the coming music, aligned to their target times and weighted by reliability. The forecasts carry the timing of upcoming beats almost as well as the actual future; with them, the generator is better synchronized with the music and concentrates more motion energy at its song's beat frequency. Commit Forcing trains continuation from the model's own committed chunks, updating both the latent history and the decoded boundary of the next chunk; it reduces low activity from 83.5% to 10.3%, the level of real dance. Three music evaluators, each validated on real G1 dance, test synchronization, song and style. On a physical Unitree G1, it dances while the music plays and comes to rest when it stops, without a fall.**

***Index Terms*—Humanoid robots, motion generation, dance generation, music-driven motion, streaming generation, diffusion models, imitation learning.**

I. Introduction

A humanoid that dances to streamed music works under constraints that offline choreography does not face. It hears only the music that has already played, yet each motion chunk is generated from the music heard up to the moment its own music begins, so that the delay from music to motion does not grow with the chunk length. And committed motion cannot be revised, so every new chunk has to continue from motion the model itself produced rather than from a clean reference. Unlike animation, every committed chunk is then executed by a whole-body tracking controller on the robot, so it must also be trackable, and generation must keep pace with the controller's clock. Under these constraints the robot must keep dancing for as long as the music plays, come to rest when it stops and dance again when it returns.

Music-to-dance generators, including long-form ones [1]–[5], receive the complete music in advance. Streaming humanoid systems hear the music causally: DiscoForcing [6] streams with diffusion forcing and retargets the generated human motion, RoboPerform [7] maps audio to a control policy, and InfiniDanceRobot [8] closes the loop between a planner and its tracker; none of them describes forecasting the coming music. Autoregressive generators trained on reference histories meet their own outputs at inference; diffusion forcing [9] and self forcing [10] address this for the history the generator conditions on, whereas a chunked robot generator also conditions on the decoded pose boundary it has committed. For a robot-native streaming generator, three questions remain: how much of the coming music a causal generator can use, how to continue reliably from its own committed chunks when both the latent history and the decoded pose boundary change, and how to show that the executed behavior follows the music rather than its own momentum.

We present AHEADance (Figs. 1 and 2), which generates humanoid reference motion chunk by chunk from streaming audio and executes it with a whole-body tracking controller. Our contributions are:

- **ListenAhead.** Each chunk is conditioned on heard music and on short-horizon forecasts of a frozen music model [11], aligned to their target times and weighted by

This work was partially supported by the Advanced Research and Invention Agency (grant number SMRB-SE01-P06) and the NVIDIA Academic Grant Program using RTX PRO 6000.

The authors are with the Department of Computer Science, University College London, London, U.K.

*Equal contribution.

†Corresponding author. chengxu.zhou@ucl.ac.uk

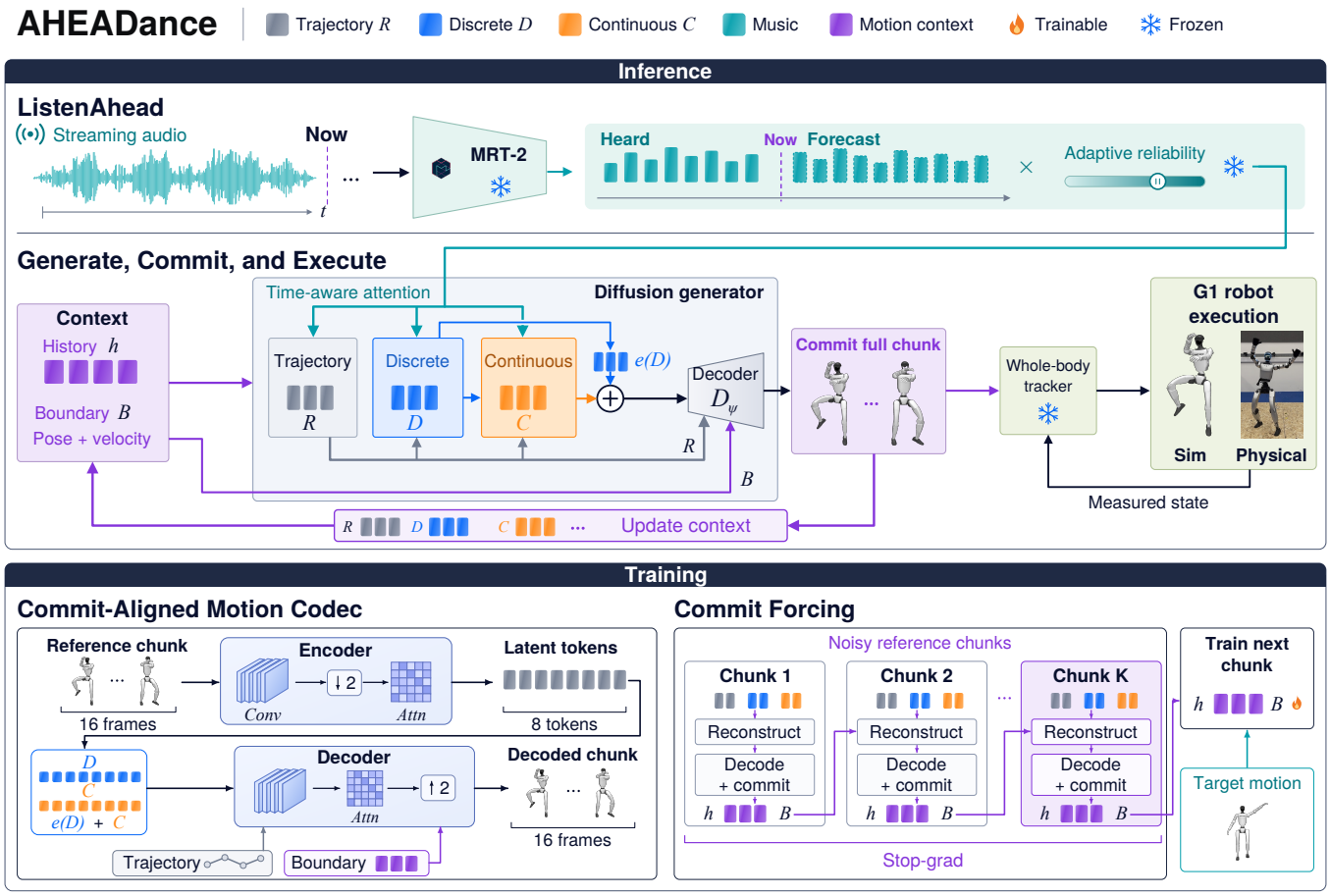


Fig. 2. Overview of AHEADance. ListenAhead builds a time-aligned music condition from heard audio and reliability-weighted forecast states. Each chunk is generated in three stages (root trajectory, discrete codes, continuous corrections), decoded once by the commit-aligned motion codec from the committed boundary, committed, and tracked by a whole-body tracker. Commit Forcing trains continuation from the model's own committed chunks. Robot renderings are schematic, except in the Physical panel.

their reliability. The forecasts carry the timing of upcoming beats almost as well as the actual future, and most of what the actual future carries about note onsets; a generator trained with them is better synchronized with the music and concentrates more motion energy at the song's beat frequency.

- **Commit Forcing.** Continuation is trained from the model's own committed outputs: noise-corrupted reference chunks are reconstructed, decoded and committed in order, and each commit updates both the latent history and the decoded reference boundary that conditions the next chunk. Low activity falls from 83.5% under reference-history training to 10.3%, the level of real dance.
- **A validated evaluation of music correspondence.** Three evaluators test humanoid dance against its music: synchronization with ChoreoSync, a detector of our own; song with MMR; and style with a style classifier. Each is validated on real G1 dance it never saw. On songs unseen in training, AHEADance reaches the highest ChoreoSync offset accuracy (28.6%) and MMR (0.187, against at most 0.046) of all compared generators.

Code, evaluation scripts and the metric-validation protocol will be released upon acceptance.

II. Related Work

A. Music-Driven and Streaming Dance

AIST++/FACT [1], Bailando [12] and Bailando++ [13] learn choreography through sequence modeling and discrete motion representations, and Danceba [14] adds gated rhythm-aware features for beat alignment. EDGE [2] and LODGE [5] extend diffusion-based synthesis to long soundtracks. ChoreoMamba uses bidirectional music conditioning for long-sequence generation [15]. FineDance [3], MEGADance [16], and SoulNet [4] further address fine-grained motion, genre-aware generation, and music–motion correspondence. Our focus is anticipatory continuation in robot coordinates as music arrives.

DiscoForcing [6] combines causal audio features with diffusion forcing for streaming character motion, followed by retargeting and tracking for humanoid execution, and shows responses to music stopping qualitatively. RoboPerform [7] instead maps audio-derived style to humanoid actions through a teacher–student policy. InfiniDanceRobot [8] streams music features into a latent diffusion-forcing planner that emits humanoid references, trains a dedicated tracker, and closes the loop by refining the planner with tracker rollouts and writing the executed state back into its history. Table I distinguishes these interfaces from offline choreography; unlike

TABLE I
INTERFACES OF MUSIC-DRIVEN METHODS

Method	Audio	Causal	Robot ref.	Own hist.	Forecast
EDGE [2]	✓	—	—	—	—
LODGE [5]	✓	—	—	—	—
DiscoForcing [6]	✓	✓	—	—	—
InfiniDance [8]	✓	✓	✓	—†	—
RoboPerform [7]	∗	✓	—	—	—
AHEADance	✓	✓	✓	✓	✓

✓: provided; —: not described. Audio: raw audio input; ∗: described in the paper; the public release takes precomputed latents. Robot ref.: the generator itself outputs the robot reference motion that a tracking controller executes (DiscoForcing generates human motion and retargets it at run time). Own hist.: continuation is trained on the model's own outputs rather than on reference history. InfiniDance: InfiniDanceRobot. †Writes the executed robot state back into the history at inference.

them, AHEADance conditions on an explicit forecast of future music representations from the frozen MRT-2 model [11].

B. Motion Representation and Generated Histories

Discrete models such as T2M-GPT [17] and MoMask [18] learn motion vocabularies, while MLD [19] uses continuous latent diffusion. DisCoRD [20] connects discrete tokens to continuous decoding, and DC-Motion [21] combines discrete structure with continuous residuals. Our formulation builds on this hybrid principle with sampled-code residual targets and explicit trajectory and boundary conditions for robot-native chunks.

Learning from generated context also has substantial precedent. Scheduled Sampling [22] and Diffusion Forcing [9] address mismatched or uncertain sequence context; DiscoForcing [6] trains on noise-corrupted reference history and down-weights distant history at inference. Self Forcing [10] trains autoregressive video diffusion through self-rollouts and sequence-level distribution matching. MotionStreamer [23] combines causal motion compression, diffusion autoregression, and Two-Forward training. Commit Forcing constructs the history and the boundary state jointly: reconstructed trajectory, codes and corrections enter committed history together, and their decoded motion sets the next reference boundary.

C. Humanoid Motion Execution

Motion imitation and learned control provide reusable execution layers. PHC [24] studies reference tracking in simulated avatars; ExBody2 [25], BeyondMimic [26], and SONIC [27] advance expressive humanoid control. HARMON [28] connects language to whole-body humanoid generation. PhysDiff [29] incorporates physical correction into denoising, while PARC [30] uses tracked motion to augment generator training. We retarget once and generate robot-native references for a fixed SONIC controller, evaluating reference quality and tracking separately.

D. Evaluating Generated Dance

Generated dance is commonly scored by Fréchet distances of kinetic and geometric motion features to real dance (FID_k, FID_g), by their diversity and by the beat alignment score (BAS) [2], [12]. Humanoid dance systems add foot sliding [6], foot contact and tracking success [8], or retrieval and tracking accuracy [7]. Learned music–motion matching scores whether motion fits its song [4], and movement descriptors characterize style [31]. We validate each music evaluator on real G1 dance it never saw, and add a detector that estimates when motion occurs relative to the music (Sec. IV-A).

III. METHOD

A. Streaming Generation with Committed Motion

At decision time τ_t, the generator has access to heard audio $a_{\leq\tau_t}$, latent history H_t, and boundary state B_t reconstructed from the last committed reference poses. Let M_t denote its music condition. Each decision generates one chunk as a root trajectory R_t, discrete codes D_t, and continuous corrections C_t (Sec. III-B).

We use a commit-aligned motion codec: encoding, generation, and commitment share the same fixed-length chunk. The codec compresses reference body motion into latent vectors and decodes each generated chunk once from its root trajectory and the preceding committed boundary. Each complete chunk is committed before the next decision, so committed motion is never decoded again.

The process has no prescribed terminal chunk: each commitment updates a bounded motion history and boundary state before the next decision. It therefore supports long-horizon generation without increasing the motion-context length with elapsed time. The training data include standing and return-to-standing segments, so the generator, without any rule, stands when the heard music falls silent and dances again when it returns. The evaluated horizons and implementation settings are given in Appendix A.

B. Commit-Aligned Motion Codec

Root trajectory represents position, height, and planar heading; the body retains full orientation and native joint angles. A new sampled code or incoming pose should change how a chunk is reconstructed without changing its body-motion target. Our formulation separates that target from the context used to decode it. A chunk encoder maps body motion to latent vectors z independently of the generated history or decoder boundary. A learned codebook contains structure embeddings e_k. For each vector,

$$D = \arg\min_k \|z - e_k\|^2, \qquad C = z - e_D. \tag{1}$$

Codes supply motion prototypes and residuals supply continuous detail, following the hybrid representation in [21]; we call this combined body representation D+C. Our decoder also receives the explicit root trajectory and current boundary:

$$x_t = \mathscr{D}\Big(e(\widehat{D}_t) + \widehat{C}_t, \widehat{R}_t, B_t\Big). \tag{2}$$

Boundary information includes pose, joint velocity, root velocity, and orientation change reconstructed from the last committed chunk, the reference the tracker executes. Body-only encoding keeps the target fixed as the preceding motion changes; the decoder adapts it to the resulting pose and velocity. The trajectory handles global placement separately from body structure.

a) Learning the codec: The shared decoder learns from structure-plus-detail, structure-only, and perturbed-residual reconstructions using robot-native pose, forward-kinematic geometry, derivative, and boundary objectives. Two consecutive chunks are decoded in sequence, as in streaming commitment; the first reconstruction supplies the next boundary. Standard quantization and commitment losses train the codebook, without predefined semantic or anatomical code labels.

b) Root, structure, then detail: The generator samples the three parts of a chunk in sequence:

$$\begin{aligned} p(R_t, D_t, C_t \mid H_t, B_t, M_t) = p(R_t \mid H_t, B_t, M_t) \\ \cdot p(D_t \mid R_t, H_t, B_t, M_t) \\ \cdot p(C_t \mid D_t, R_t, H_t, B_t, M_t). \end{aligned} \quad (3)$$

Continuous diffusion first samples the trajectory, which fixes global placement; categorical diffusion then samples the codes given the trajectory, and conditional continuous diffusion samples the corrections given both. All three stages share the history, boundary, and music condition. Target positions within a chunk are modeled jointly, while committed history remains fixed.

c) Detail conditioned on sampled structure: The reference residual is tied to the reference code. Pairing it with a different sampled code would shift the combined latent away from the intended body target. We instead rebase the continuous target around the code actually selected:

$$C_{\text{target}} = z^* - e(\widehat{D}), \qquad e(\widehat{D}) + C_{\text{target}} = z^*. \quad (4)$$

Thus different code samples retain the same combined latent target z^* while inducing different correction targets. This is distinct from learning a residual around a fixed reference code. Together with boundary-independent encoding, it separates the intended body motion from variations in sampled structure and preceding motion. Training normalizes residuals with frozen training-set statistics.

d) Information access at the commitment scale: Because the codec chunk is the commitment unit, the decoder attends jointly over the chunk's whole generated plan, coordinating its entrance and exit, before the chunk is committed. This reads no unheard waveform: causality is imposed at the music decision boundary, not on plan variables already generated inside the chunk. Across chunks, the committed pose and velocity and cross-boundary derivative supervision provide continuity.

C. ListenAhead Music Conditioning

The chunk generated at decision time τ_t accompanies music that has not yet been heard. ListenAhead supplies that music as a forecast. A frozen MRT-2 model predicts future hidden states from the observed audio prefix [11]: 14 states at 25 Hz whose target times reach about 0.5 s past the audio cutoff τ_t, where the chunk begins, and so span nearly all of its 0.533 s; each is the mean of eight sampled continuations in hidden space. The forecast is computed once per decision and held fixed while the chunk is sampled; Sec. IV-C measures what it carries. ListenAhead turns it into a motion condition with two explicit attributes: when a state is expected and how much the generator should rely on it.

The music condition M_t has three parts: the heard music, the forecasts, and a MusicCoCa style embedding [32] of the trailing 10 s. Heard music enters through gated cross-attention over its states, style through a scale and shift of the hidden states, and the forecasts through the time-aware attention below.

a) When a forecast is expected: Predicted states retain their physical time rather than becoming an unordered future summary. For target motion position i and forecast state j, let Δ_{ij} be the difference between their times. Attention combines a content score with a timing term $\beta(\Delta_{ij})$, a per-head linear map of sinusoidal features of Δ_{ij} at 32 log-spaced frequencies:

$$w_{ij} = \text{softmax}_j \left(q_i^\top k_j / \sqrt{d} + \beta(\Delta_{ij}) \right), \quad (5)$$

$$u_i = \sum_j w_{ij} \rho_j v_j, \quad (6)$$

where invalid states are masked and ρ_j is prediction reliability. The timing term matches upcoming musical context to a motion position.

b) How much to rely on a forecast: Forecasts are not equally good. The reliability $\rho_j \in [0,1]$ estimates how much better forecast j is than holding the last heard state. It is a ridge regression on causal features: the lead time, how far the forecast departs from the last heard state, the recent activity of the music, and the realized skill of the two preceding forecasts. It is fitted once on the training split and uses the audio prefix only. Reliability scales the forecast's contribution after attention normalization. This placement lets the entire forecast pathway weaken when predictions are uncertain, rather than redistributing a unit attention mass among unreliable states.

c) Entering training at zero: The forecast pathway is zero-initialized: its output projection and timing term start at zero, so the forecasts contribute nothing at the start of training and enter only as far as they reduce the loss. Masking all forecast states and setting $\rho_j = 0$ removes the pathway; this is the w/o-future model of Sec. IV-C. The same music interface conditions trajectory, discrete-code, and continuous-correction generation.

D. Commit Forcing

During generation, latent history and the incoming boundary both arise from earlier model output. Replacing recorded latents with generated ones while retaining a recorded boundary would describe two different motion histories. Commit Forcing constructs both conditions from the same decoded commitments.

After reference-history warmup, the current generator reconstructs trajectory, codes, and continuous corrections from noise-corrupted reference chunks with one denoising step (Appendix A-b). Starting from the context at the beginning of the history window, it processes chunks $j = 1, \ldots, K$ in sequence. Each reconstruction is decoded and committed before the next one:

$$(H_j, B_j) = \mathscr{U}(H_{j-1}, B_{j-1}, \widehat{R}_j, \widehat{D}_j, \widehat{C}_j). \quad (7)$$

Here $\mathscr{U}$ forms H_j by appending generated latent components and reconstructs B_j from their decoded terminal poses. The next history reconstruction receives this updated pair, so changes in generated motion reach both the latent context

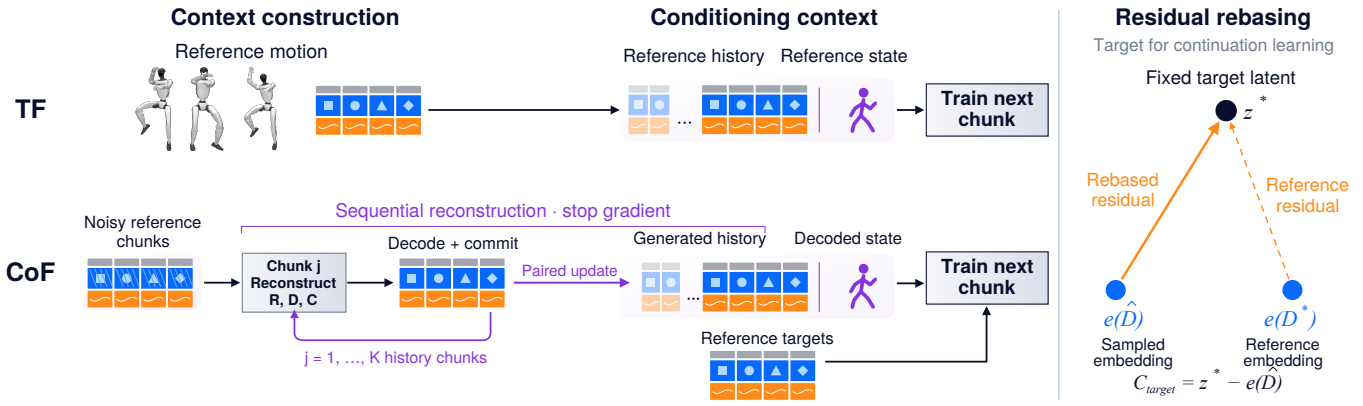


Fig. 3. **Commit-consistent continuation training.** Reference-history training supplies recorded contexts. Commit Forcing reconstructs a sequence of K perturbed reference chunks, decodes and commits each complete chunk, and carries the updated context into the next reconstruction. Each update pairs generated history with the boundary state of that same decoded motion. This history construction stops gradients; the current continuation is supervised with reference targets. The inset uses the shared code-relative target rule of Eq. 4.

and the starting state. After K updates, (H_K, B_K) conditions continuation learning.

History reconstruction stops gradients; code prediction, continuous denoising, and decoded-motion objectives supervise the next chunk. The residual target rule in Sec. III-B is shared by history reconstruction and continuation supervision.

E. Robot Execution

SONIC [27] tracks the generated motion in simulation. The interface converts joint, orientation, and timing conventions. The generator updates its history and boundary state from committed motion; SONIC uses measured state for feedback control. The music scores in Sec. IV are computed on the generated reference motion, before execution.

IV. Experiments

Table II compares generators, Table III isolates ListenAhead and Commit Forcing, and Table IV validates the evaluators on real G1 dance.

A. Setup and Metrics

We use dance motion from six public datasets retargeted to our test platform, the Unitree G1 humanoid: AIOZ-GDANCE [33], FineDance [3], CoMPAS3D [34], AIST++ [1], ChoreoMaster [35] and OpenDance [36], as released in OMG-Data [37] with that release's official splits (61.7 h of training motion).

Evaluation sets. Ablations use 721 sequences (224 songs) of at least 8 s from the test and validation splits, none of which ChoreoSync was trained on: AIOZ-GDANCE 506, FineDance 103, AIST++ 95 and CoMPAS3D 17. Generators are compared on 42 of these songs: by audio fingerprint, their audio occurs neither in any generator's training data nor in that of ChoreoSync or MMR.

Music scores. Three evaluators test how generated dance corresponds to its music: in synchronization, in song and in style.

TABLE II
Comparison on 42 Unseen Songs

Method	$FID_k \downarrow$	Div_k	Sync↑	MMR↑	Tempo↑
Real dance	–	6.22	69.0	0.245	80.2
Online (no future music)					
AHEADance	**18.8**	2.22	**28.6**	**0.187**	57.4
Bailando++, strict	65.2	0.59	11.9	−0.010	48.1
Danceba, strict	44.8	4.52	9.5	−0.003	46.6
MEGADance, strict	23.7	3.77	11.9	−0.011	49.4
Offline					
EDGE	27.6	2.22	27.8	0.046	**58.8**
Lodge	44.1	1.70	14.3	0.044	46.2
MEGADance	47.1	2.46	10.3	0.008	46.8
Bailando++	77.8	0.27	4.8	−0.007	50.2
Danceba	61.0	1.79	11.9	0.007	53.4

42 AIOZ-GDANCE test songs whose audio occurs in no model's training data. All motion is generated G1 reference motion, before execution; every model of a song is scored on the same frames. Sync: ChoreoSync offset accuracy (%, chance 11.1). Other scores as in Table III; FID_k against real dance of these songs. No Style column: our style classifiers cover only the FineDance and AIST++ style labels; Appendix A-h compares style on FineDance songs. Bailando++ and Danceba: one sample per song; others three. Online, strict: the model runs on causal input. Baseline details in Appendix A-g. Bold: best generator in the column.

Synchronization. ChoreoSync, our contrastive motion–music detector trained on G1 dance, estimates the music offset at which a clip's motion and music agree best; offset accuracy, the share of clips whose estimate lies within 0.2 s of the true alignment, measures whether movement is aligned in time with its own music (Appendix A-c).

Song. MMR, the music–motion retrieval model of SoulNet [4] retrained on G1 motion, scores the margin of a motion window's similarity to its own music over other songs' music at the same time, testing whether the window matches its song; every sequence is scored only by evaluators not trained on it (Appendix A-c).

Style. A classifier over movement descriptors [31] labels FineDance dances Classic, Folk or Street, and a second labels AIST++ dances by their ten genres; style accuracy is the share

TABLE III
TRAINING ABLATIONS ON 721 HELD-OUT SEQUENCES

	Music				Continuation		Motion distribution		
Setting	ChoreoSync↑	MMR↑	BAS↑	Tempo↑	Low act.↓	Span↑	FID_k ↓	FID_g ↓	Div_k
Real dance	45.2	0.184	0.247	71.3	9.7	0.922	–	–	6.28
AHEADance	**24.2**	**0.126**	0.262	**56.8**	**10.3**	**0.732**	**16.7**	**46.3**	2.94
w/o future channel	17.4	0.054	0.268	51.3	17.5	0.604	21.3	74.1	3.11
w/o forecasts (last heard state)	15.8	0.047	0.265	48.3	10.8	0.636	23.2	71.1	2.55
Teacher Forcing	18.4	0.023	0.238	47.6	83.5	0.492	64.6	140.6	1.53
CoF w/o boundary closure	18.5	0.029	0.294	53.3	72.5	0.509	45.1	123.8	1.77

Each row is a separately trained model (three sampling seeds); test and validation sequences of at least 8 s, up to 32.8 s (224 songs). w/o forecasts: the future channel holds the last heard music state. ChoreoSync: offset estimated within 0.2 s of the true alignment (%, chance 11.1). MMR: margin of a motion window's similarity to its own music over other songs' music at the same time. BAS: beat alignment score. Tempo: tempo salience (%, chance 50). Low act.: 4 s windows below the 10th percentile of training motion (%). Span: joint range (rad). FID_k, FID_g and Div_k (descriptive): 5 s windows, per dataset against the real dance of these sequences (Appendix A-e). Bold: best generator in the column; BAS is not ranked.

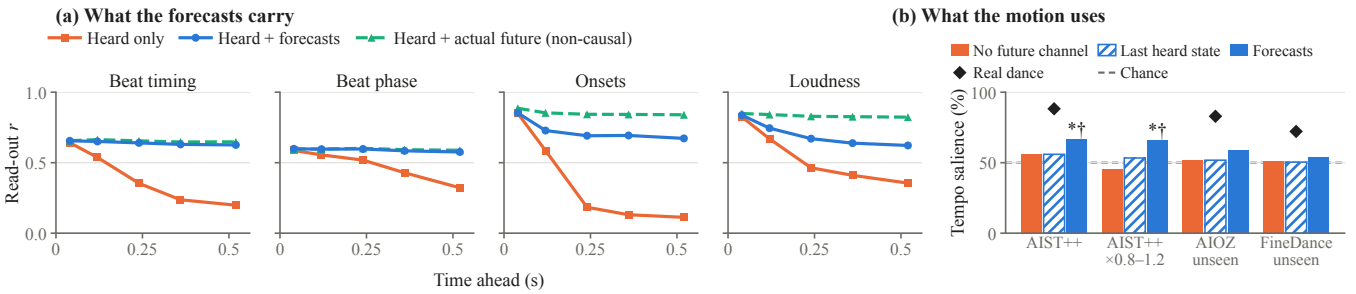


Fig. 4. ListenAhead. (a) What the forecasts carry: cross-validated linear read-outs (Pearson r on held-out songs; five song folds, 235 songs) of music descriptors, by how far ahead, from MRT-2 states of the heard music, the heard music with the forecasts, or the heard music with the actual future (a non-causal upper bound). (b) What the motion uses: tempo salience of three generators (without the future channel, with it holding the last heard music state, and with the forecasts) on AIST++ music (60 pieces), ten of these pieces time-stretched to five speeds between 0.8× and 1.2×, and songs unseen in training (AIOZ-GDANCE 42, FineDance 30). *, †: the 95% CI of the paired difference over songs from the generator without the future channel (*) or with the last heard state (†) excludes zero (t interval for the ten time-stretched pieces). Diamonds: real dance of the same songs (51 AIST++ pieces and 27 FineDance songs; none for time-stretched music); dashed line: chance.

TABLE IV
SCORE VALIDATION ON REAL G1 DANCE

Score	AIOZ	FineDance	AIST++	CoMPAS3D	All	Chance
Sequences	506	103	95	17	721	
ChoreoSync (%)	46.2	26.2	53.7	82.4	45.2	11.1
MMR (%)	68.4	68.7	67.6	80.9	68.6	50
Tempo (%)	71.2	68.1	79.3	45.9	71.3	50
Style (%)	–	84.2	65.3	–	–	33 / 10
BAS, own music	0.247	0.216	0.283	0.212	0.247	–
BAS, other songs	0.244	0.214	0.250	0.250	0.241	–

Test and validation sequences that the evaluators were not trained on. MMR: share of comparisons in which a window's own music scores above another song's music at the same time. Style: 76 labelled FineDance sequences and the 95 AIST++ sequences, classifiers that never saw the song; chance 33% on FineDance (three classes), 10% on AIST++ (ten genres). Tempo salience on CoMPAS3D stays at chance.

of samples classified as their song's style, each song scored by a classifier fitted without it. Style following is the rate at which a dance is classified as its own song's style minus the mean rate at which it is classified as the style of each other song, so a dance whose style ignores its music scores zero.

The evaluators are not used to train the generator. We also report the common beat alignment score (BAS) [12].

Tempo salience. A rule-based score of beat-frequency energy: the share of comparisons in which a dance concentrates more motion energy at its own song's beat frequency than at that of a rival song whose tempo differs by at least 8%.

Validation. Each evaluator is validated on real G1 dance (Table IV; Appendix A-c). ChoreoSync estimates the offset within 0.2 s for 45.2% [39.5, 50.9] of the 721 sequences (chance 11.1%), and MMR prefers a dance's own music in 68.6% [65.9, 71.4] of comparisons (chance 50%). BAS does not distinguish real dance on its own music from real dance on other songs' music (0.247 vs. 0.241).

Motion. Joint span and low activity characterize sustained movement. Motion distributions are compared with the kinetic and geometric Fréchet distances (FID_k, FID_g) and kinetic diversity (Div_k) [12]; windows and normalization are given in Appendix A-e. Appendix A-f adds FID_{LMA} and Laban movement qualities. Paired differences use the compact notation *estimate [95% confidence interval]*, with the complete model minus the compared model.

B. Comparison with Other Generators

We compare AHEADance with EDGE [2], Lodge [5], MEGADance [16], Bailando++ [13] and Danceba [14]. To compare them under the same causal access, we also run Bailando++, Danceba and MEGADance strictly online, with music features recomputed at each step from the heard audio only and token decoders that see no future tokens (Appendix A-g).

On 42 songs unseen in training, AHEADance has the lowest FID_k and the highest MMR of all generators (Table II): FID_k 18.8 against at least 23.7, and MMR 0.187 against at most 0.046, with the 95% CI of every difference excluding zero. Its ChoreoSync offset accuracy, 28.6%, is also the highest. RoboPerform's release takes music latents precomputed for its training clips and cannot be given other music, and DiscoForcing's weights are not released (our reproduction is reported in Appendix A-g), so we compare their interfaces (Table I). On 23 FineDance songs that AHEADance never heard, each scored by a classifier fitted without it, its style following is 32.6 [22.5, 40.9] points (real dance 47.4): higher than that of every generator given no genre and every strictly online variant, not significantly different from that of Lodge, and lower than that of offline MEGADance, both of which receive each song's genre and were trained on FineDance (Appendix A-h).

C. Music Alignment with ListenAhead

Trained without the future channel, the generator's ChoreoSync offset accuracy is 6.8 [2.4, 11.2] points lower (17.4% vs. 24.2%) and its tempo salience 5.5 [2.3, 8.7] points lower (51.3% vs. 56.8%; Table III). The future channel also lets tempo follow the music when it is played 0.8–1.2× faster or slower: tempo salience rises from 45.0% to 65.7% (20.7 [14.3, 27.1]; higher for all ten pieces; Fig. 4b). With the forecast slots filled by the last heard music state instead, the gain disappears at the original tempo: tempo salience on the 60 AIST++ pieces is 10.6 [5.8, 15.5] points lower than with the forecasts and no higher than without the future channel, and ChoreoSync offset accuracy is 8.4 [3.3, 13.5] points lower (15.8% vs. 24.2%; Table III). At 0.8–1.2× it is 12.3 [6.4, 18.2] points lower, keeping 40% of the gain, so the gain comes mainly from what the forecasts predict rather than from an extra time-aligned input.

What the forecasts encode. We read music descriptors from MRT-2 states with cross-validated linear read-outs (PCA and ridge regression) on 235 songs from FineDance, AIST++, AIOZ-GDANCE, PhantomDance and 16 other tracks (Fig. 4a). The forecasts anticipate rhythm: on average over 0.04–0.52 s ahead, they provide 95% of the gain that the actual future gives for the timing of upcoming beats, 96% for beat phase and 74% for note onsets, and help least for changes in loudness. A causal beat tracker that extrapolates the heard beats recovers 46% of the forecasts' gain for beat timing, 58% for beat phase and 12% for onsets. At 0.52 s ahead, beat timing is read with $r = 0.63$ from the forecasts (actual future 0.65) and $r = 0.20$ from the heard music alone.

D. Sustained Motion with Commit Forcing

Commit Forcing sustains movement relative to reference-history training (Table III). Low activity falls from 83.5% to 10.3%, the level of real dance (9.7%; −73.1 [−76.2, −69.9] points). Later in continuation the gap widens: from 8.8 to 32.8 s, on the 284 sequences that reach 32.8 s, low activity falls from 98.5% to 8.4%. Commit Forcing also moves motion closer to real dance: FID_k falls from 64.6 to 16.7 (−47.9 [−53.6, −42.6]) and FID_g from 140.6 to 46.3 (−94.3 [−104.7, −87.6]). With the decoded reference boundary held at the reference (latent history still self-generated), low activity rises to 72.5% (−62.1 [−65.2, −59.0] points for the complete model) and FID_k to 45.1 (−28.4 [−33.7, −24.3]), so once the history is self-generated, the decoded boundary must be generated too.

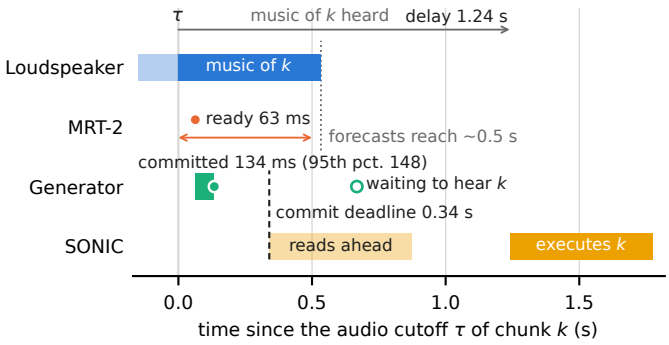


Fig. 5. One chunk on the robot, relative to its audio cutoff τ: medians over the 12 physical runs. Dashed: commit deadline (1.24 s delay minus the 0.9 s preview); open circle: commit time if each chunk's music were heard before generating it.

E. Execution on the Robot

AHEADance generates reference motion faster than playback: on the robot, a 16-frame chunk (0.533 s at 30 Hz) is committed 0.13 s after its audio cutoff (0.15 s at the 95th percentile), forecasting included (Fig. 5; Appendix A-n). Because the SONIC tracker reads the reference about 0.9 s ahead, we execute on the robot with a 1.24 s delay; waiting to hear each chunk's music before generating it would add the chunk length, 0.53 s. With the official low-latency SONIC (0.32 s preview), the same generator runs all 8 programmes in real time in simulation with a 0.8 s delay, which waiting to hear each chunk's music could not meet (Appendix A-k). SONIC executes the generated references in simulation successfully (no reset, pelvis height within 0.25 m) in 99.5% of 369 trials on the test split.

Dancing to music includes coming to rest when it stops (Sec. III). We interrupt songs with 60 s of silence in 28 programmes, each paired with a control that continues the music from the same generator state (Appendix A-k). AHEADance stops and dances again in every programme; executed by SONIC in simulation, it is near standing a median 2.32 s after the music stops on the 16 songs from the test split. Of the other generators, all trained on dance alone, Bailando++ stops and dances again in 3 programmes, Danceba in 2, and EDGE, Lodge, MEGADance and the strictly online variants in none. On a physical Unitree G1, the music is streamed from audio files and played through a loudspeaker as it arrives, so the robot moves about 1.2 s behind the music the audience hears. Near-standing followed each of the 20 music stops and never occurred while the robot danced, and sustained dancing followed each of the 20 starts, a median 1.75 s and 0.85 s later in the stream (about 3.0 s and 2.1 s for the audience); no run fell, and joints track the reference with a median error of 5.5° (Fig. 6; supplementary video; Appendix A-l).

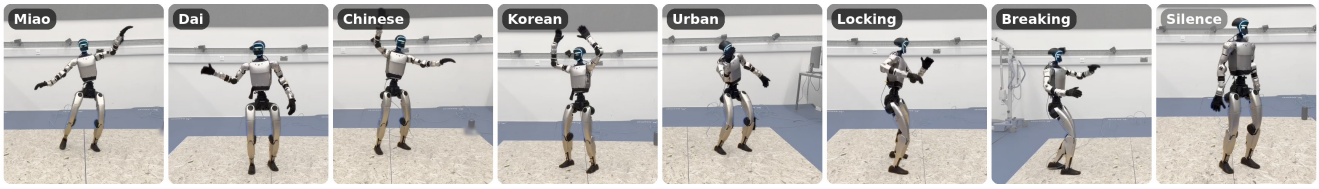


Fig. 6. The physical Unitree G1 dancing to songs of seven styles and standing after the music stops (backgrounds lightened).

V. Discussion

The forecasts carry the timing of the next beats almost as well as the actual future, and most of what the actual future carries about note onsets; the generator converts them into better synchronization and more motion energy at the music's beat frequency. Turning anticipated beats into the timing of movement accents is the most direct next step (Sec. IV-C), followed by a perceptual study of how audiences judge the dance.

AHEADance's kinetic diversity still remains below that of real dance. Forecasts anticipate regular structure such as beats, but not an abrupt change such as a sudden stop, to which the robot responds afterwards (about 1 s in simulation). The music scores are measured on the generated references, which a general-purpose tracker executes after a fixed delivery delay; a tracker trained for dance could preserve more of their beat-level detail. The generator does not receive the measured robot state, and it runs off-board on a workstation; feeding the state back and running on the robot are left to future work.

VI. Conclusion

AHEADance generates robot-native dance from streaming audio under causal music access and irrevocable commitments. A future-aligned music channel anticipates beat timing and improves synchronization and beat-frequency motion over a generator trained without it, a gain that the last heard music state alone does not provide, and training continuation from the model's own committed chunks keeps low activity at the level of real dance. ChoreoSync and MMR, validated on real G1 dance, show that AHEADance follows unseen songs more closely than other generators, and on FineDance songs it never heard, its style follows the music more closely than that of any generator not given the genre.

References

[1] R. Li, S. Yang, D. A. Ross, *et al.*, "AI Choreographer: Music conditioned 3D dance generation with AIST++," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 401–13 412.

[2] J. Tseng, R. Castellon, and C. K. Liu, "EDGE: Editable dance generation from music," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 448–458.

[3] R. Li, J. Zhao, Y. Zhang, *et al.*, "FineDance: A fine-grained choreography dataset for 3D full body dance generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 234–10 243.

[4] X. Li, R. Li, S. Fang, *et al.*, "Music-aligned holistic 3D dance generation via hierarchical motion modeling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 14 420–14 430. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2025/html/Li_Music-Aligned_Holistic_3D_Dance_Generation_via_Hierarchical_Motion_Modeling_ICCV_2025_paper.html

[5] R. Li, Y. Zhang, Y. Zhang, *et al.*, "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, pp. 1524–1534.

[6] K. Ji, B. Qian, B. Wu, *et al.*, "DiscoForcing: A unified framework for real-time audio-driven character control with diffusion forcing," in *Proceedings of the 43rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 306, 2026, pp. 51 514–51 544. [Online]. Available: https://proceedings.mlr.press/v306/ji26h.html

[7] Z. Li, C. Chi, Y. Wei, *et al.*, "Do you have freestyle? expressive humanoid locomotion via audio control," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 956–965. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2026/papers/Li_Do_You_Have_Freestyle_Expressive_Humanoid_Locomotion_via_Audio_Control_CVPR_2026_paper.pdf

[8] J. Zhang, C. Li, C. Lin, *et al.*, "InfiniDanceRobot: A streaming and closed-loop music-driven dance generation for humanoid robot," in *Conference on Robot Learning (CoRL)*, 2026. [Online]. Available: https://openreview.net/forum?id=93leGzRaPV

[9] B. Chen, D. Martí Monsó, Y. Du, *et al.*, "Diffusion forcing: Next-token prediction meets full-sequence diffusion," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 24 081–24 125.

[10] X. Huang, Z. Li, G. He, *et al.*, "Self forcing: Bridging the train-test gap in autoregressive video diffusion," in *Advances in Neural Information Processing Systems*, vol. 38, 2025. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2025/hash/f4823f831af67a3ef15e41a85434422a-Abstract-Conference.html

[11] Magenta Team, "Magenta RealTime 2: Open & local live music models," Magenta blog, June 2026. [Online]. Available: https://magenta.withgoogle.com/magenta-realtime-2

[12] L. Siyao, W. Yu, T. Gu, *et al.*, "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, pp. 11 050–11 059.

[13] L. Siyao, W. Yu, T. Gu, *et al.*, "Bailando++: 3D dance GPT with choreographic memory," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 12, pp. 14 192–14 207, 2023.

[14] C. Fan, J. Guan, X. Zhao, *et al.*, "Align your rhythm: Generating highly aligned dance poses with gating-enhanced rhythm-aware feature representation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 13 193–13 202.

[15] V. Tan, H. Kim, S. Choi, *et al.*, "ChoreoMamba: Long-sequence dance generation with bidirectional mamba," *Journal of the Korea Computer Graphics Society*, vol. 32, no. 4, pp. 1–12, 2026. [Online]. Available: https://www.kci.go.kr/kciportal/ci/sereArticleSearch/ciSereArtiView.kci?sereArticleSearchBean.artiId=ART003371445

[16] K. Yang, X. Tang, Z. Peng, *et al.*, "MEGADance: Mixture-of-experts architecture for genre-aware 3D dance generation," in *Advances in Neural Information Processing Systems*, vol. 38, 2025. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2025/hash/102bd3c1076dfe98043b19340ac44e08-Abstract-Conference.html

[17] J. Zhang, Y. Zhang, X. Cun, *et al.*, "Generating human motion from textual descriptions with discrete representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 730–14 740.

[18] C. Guo, Y. Mu, M. G. Javed, *et al.*, "MoMask: Generative masked modeling of 3D human motions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1900–1910.

[19] X. Chen, B. Jiang, W. Liu, *et al.*, "Executing your commands via motion diffusion in latent space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 18 000–18 010.

[20] J. Cho, J. Kim, J. Kim, *et al.*, "DisCoRD: Discrete tokens to continuous motion via rectified flow decoding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 14 602–14 612.
[21] H. Wang, X. Chen, J. Zhang, *et al.*, "DC-Motion: Decoupling structure and details via discrete–continuous tokens for human motion generation," *arXiv preprint arXiv:2606.14721*, 2026. [Online]. Available: https://arxiv.org/abs/2606.14721
[22] S. Bengio, O. Vinyals, N. Jaitly, *et al.*, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, vol. 28, 2015, pp. 1171–1179.
[23] L. Xiao, S. Lu, H. Pi, *et al.*, "MotionStreamer: Streaming motion generation via diffusion-based autoregressive model in causal latent space," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 10 086–10 096.
[24] Z. Luo, J. Cao, A. Winkler, *et al.*, "Perpetual humanoid control for real-time simulated avatars," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 895–10 904. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/papers/Luo_Perpetual_Humanoid_Control_for_Real-time_Simulated_Avatars_ICCV_2023_paper.pdf
[25] M. Ji, X. Peng, F. Liu, *et al.*, "ExBody2: Advanced expressive humanoid whole-body control," *arXiv preprint arXiv:2412.13196*, 2024. [Online]. Available: https://arxiv.org/abs/2412.13196
[26] Q. Liao, T. E. Truong, X. Huang, *et al.*, "BeyondMimic: From motion tracking to versatile humanoid control via guided diffusion," *arXiv preprint arXiv:2508.08241*, 2025. [Online]. Available: https://arxiv.org/abs/2508.08241
[27] Z. Luo, Y. Yuan, T. Wang, *et al.*, "SONIC: Supersizing motion tracking for natural humanoid whole-body control," *Science Robotics*, vol. 11, no. 117, p. eaed4592, 2026. [Online]. Available: https://arxiv.org/abs/2511.07820
[28] Z. Jiang, Y. Xie, J. Li, *et al.*, "Harmon: Whole-body motion generation of humanoid robots from language descriptions," in *Proceedings of the 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 270. PMLR, 2025, pp. 3015–3026. [Online]. Available: https://proceedings.mlr.press/v270/jiang25b.html
[29] Y. Yuan, J. Song, U. Iqbal, *et al.*, "PhysDiff: Physics-guided human motion diffusion model," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 16 010–16 021. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/html/Yuan_PhysDiff_Physics-Guided_Human_Motion_Diffusion_Model_ICCV_2023_paper.html
[30] M. Xu, Y. Shi, K. Yin, *et al.*, "PARC: Physics-based augmentation with reinforcement learning for character controllers," in *ACM SIGGRAPH 2025 Conference Papers*, 2025.
[31] M. Turab, P. Colantoni, D. Muselet, *et al.*, "Dance style recognition using Laban movement analysis," in *International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, ser. Lecture Notes in Computer Science. Springer, 2025, pp. 564–575.
[32] A. Caillon, B. McWilliams, C. Tarakajian, *et al.*, "Live music models," *arXiv preprint arXiv:2508.04651*, 2025.
[33] N. Le, T. Pham, T. Do, *et al.*, "Music-driven group choreography," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8673–8682.
[34] B. Burkanova, Y. Etesam, P. J. Yazdian, *et al.*, "CoMPAS3D: A dataset and benchmark for interactive motion," *arXiv preprint arXiv:2507.19684*, 2025.
[35] K. Chen, Z. Tan, J. Lei, *et al.*, "ChoreoMaster: Choreography-oriented music-driven dance synthesis," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 1–13, 2021.
[36] J. Zhang, Z. Kang, L. Liu, *et al.*, "OpenDance: Multimodal controllable 3D dance generation with large-scale internet data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026, pp. 28 860–28 870. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2026/html/Zhang_OpenDance_Multimodal_Controllable_3D_Dance_Generation_with_Large-scale_Internet_Data_CVPR_2026_paper.html
[37] S. Huang, K.-Y. Lee, D. Qiao, *et al.*, "OMG: Omni-modal motion generation for generalist humanoid control," *arXiv preprint arXiv:2606.10340*, 2026.
[38] H. Zhu, Y. Zhou, H. Chen, *et al.*, "MuQ: Self-supervised music representation learning with mel residual vector quantization," *arXiv preprint arXiv:2501.01108*, 2025. [Online]. Available: https://arxiv.org/abs/2501.01108
[39] J. P. Araujo, Y. Ze, P. Xu, *et al.*, "Retargeting matters: General motion retargeting for humanoid motion tracking," *arXiv preprint arXiv:2510.02252*, 2025.
[40] M. Petrovich, M. J. Black, and G. Varol, "TEMOS: Generating diverse human motions from textual descriptions," in *Proceedings of the European Conference on Computer Vision*, 2022, pp. 480–497.
[41] M. Petrovich, M. J. Black, and G. Varol, "TMR: Text-to-motion retrieval using contrastive 3D human motion synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9488–9497.
[42] P. Dhariwal, H. Jun, C. Payne, *et al.*, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020. [Online]. Available: https://arxiv.org/abs/2005.00341

APPENDIX A
SUPPLEMENTARY RESULTS AND NOTES

a) Evaluation settings: The six datasets follow the official splits of the OMG-Data release [37] (https://huggingface.co/datasets/THU-MARS/OMG-Data, revision 6e0dfbc): 61.7 h of training, 6.9 h of validation and 7.5 h of test motion. Training motion comprises AIOZ-GDANCE 48.5 h, FineDance 5.5 h, CoMPAS3D 4.5 h, AIST++ 2.4 h, ChoreoMaster 0.9 h and OpenDance 0.04 h. For FineDance, this split differs from the dataset's own test list: the audio of 15 of its 18 test songs occurs in the training data, which applies equally to every row of Table III. The 721 sequences are the test and validation sequences of at least 8 s, none of which was used to train ChoreoSync or the MMR evaluator that scores it. By audio fingerprint, the music of 42 of them occurs in ChoreoSync's training data and that of 101 in MMR's, through other dances to the same songs; every row of Table III is scored on the same sequences. Without these sequences, the ChoreoSync differences of Sec. IV-C are 7.0 $[2.4, 11.6]$ and 8.7 $[3.3, 13.9]$ points, and AHEADance's MMR stays above that of every ablation in Table III, each 95% CI excluding zero. The 721 sequences run up to 32.8 s, with three sampling seeds (1234, 2345, 3456) and ten sampling steps. Low activity from 8.8 to 32.8 s uses the 284 that reach 32.8 s (113 songs). The 42 AIOZ-GDANCE songs are a subset of these sequences, screened with a landmark audio fingerprint against AHEADance's training data; the screen removed two of 44 songs. Every model of a song is cut to the same frames, inside the song's real audio.

b) Implementation: AHEADance is trained from scratch (seed 1234) with a static standing initial boundary. The codec (10.7 M parameters) encodes two frames per latent vector (16 dimensions; codebook of 512 entries). The generator (619 M parameters; width 768, 12 layers with 12 heads, 9 diffusion layers) conditions on 64 history tokens (128 frames, 4.27 s) and generates eight tokens (one 16-frame chunk) per decision with ten sampling steps per stage. MRT-2 (small) samples its eight continuations at temperature 1.3 and top-k 40. The reliability regression is clipped to $[0, 1]$. Commit Forcing reconstructs and commits the $K = 8$ chunks (4.27 s) preceding each training window. Training uses reference histories first and then switches to reconstructed histories, without a gradual mixing schedule. Each source window draws one shifted logit-normal noise strength (shift 0.6), shared across its successive chunks; each chunk is reconstructed with one denoising step, and the history is built without gradient propagation. The generator is trained on four GPUs with 1,536 target chunks per update; generation runs off-board on a workstation with separate GPUs for the music model and the generator.

c) Score validation: A music–motion score supports claims about musical correspondence only if it prefers a dance's own music to other music. Our MMR is checked this way on sequences its scoring evaluator was not trained on and, unlike SoulNet's evaluator [4], is never used to train the generator.

ChoreoSync has two Transformer encoders, f_m and f_a, that map a 4 s window of G1 motion features and a 4 s window of frozen MuQ music features [38] to unit embeddings. A symmetric contrastive loss takes the time-aligned music window as the positive and, as negatives, windows of the same song shifted by at least 0.15 s plus other songs in the batch, so that the embedding must encode when movement happens relative to the music rather than which song plays. A clip is scored by the offset curve

$$c(\delta) = \frac{1}{|\mathscr{W}|} \sum_{t \in \mathscr{W}} f_m(x_t)^\top f_a(a_{t+\delta}), \qquad \delta \in [-2, 2]\,\mathrm{s}, \quad (8)$$

over motion windows x_t starting every 1 s and music windows $a_{t+\delta}$ shifted in 0.05 s steps. The clip's offset estimate is the peak $\hat{\delta} = \arg\max_\delta c(\delta)$. ChoreoSync is trained on the training splits of the same datasets as the generator and on real dance retargeted with GMR [39], so that one detector scores both retargeting routes. On generated motion, ChoreoSync follows known shifts: shifting AHEADance's dance against its music by ± 0.25, ± 0.5 or ± 1 s (679 sequences of 207 songs it never heard) moves the estimated offset by the same amount (median error at most 0.10 s), and accuracy against the unshifted alignment falls from 24.0% to 8.6–13.7% (chance 11.1%) for shifts of 0.5 s or more.

MMR follows the music–motion retrieval module of SoulNet [4], a contrastive encoder–decoder in the TEMOS/TMR style [40], [41] over Jukebox features [42], trained here on G1 motion. It has three evaluators, each trained on two thirds of the FineDance, AIST++ and PhantomDance songs and on all training clips of AIOZ-GDANCE and CoMPAS3D of at least 5 s. FineDance and AIST++ sequences are scored by the evaluator that holds them out, AIOZ-GDANCE and CoMPAS3D sequences by the mean of the three. MMR's own-music rates on real dance have 95% CIs [65.4, 71.3], [63.5, 73.9], [54.9, 79.4] and [74.4, 87.4]% on AIOZ-GDANCE, FineDance, AIST++ and CoMPAS3D. Every style score, of real and of generated dance, uses a classifier refitted without the scored song's fold and without songs that share its audio. BAS on real dance, own music minus other songs' music, is $0.006\ [-0.002, 0.014]$ on the 721 sequences.

d) Chunk representation: D+C is more precise than discrete-only generation and moves more than continuous-only generation (Table V). Codes alone cannot place a pose precisely: on the 123 sequences of the test split, decoding encoded motion chunk by chunk from its own decoded boundary, as in streaming, gives 7.2° mean joint error from codes alone, against 2.9° with residuals (higher for 122 of 123 sequences). In simulation on the same test split, SONIC tracks discrete-only generation with $1.24\ [0.97, 1.50]^\circ$ more joint error and $4.3\ [2.5, 5.6]$ mm more position error than D+C (369 trials per model). Continuous-only generation is tracked as accurately as D+C, and its low activity is 34.6% against 10.3%.

e) Metrics: MMR takes its margin over the mean of other songs' music, in windows at 5, 10, …, 25 s; it is a score, not an accuracy percentage. BAS uses nearest music-to-motion beat matching on G1 keypoint speeds with a three-frame Gaussian tolerance at 30 Hz: music beats come from a beat tracker, motion beats from local minima of the smoothed mean keypoint speed. Tempo salience runs from 1 s to the end of a clip. Joint span is the mean 95th–5th percentile joint range; low activity counts 4-second windows below the 10th percentile of training-motion activity, so real dance is near 10%. FID_k, FID_g and Div_k use 5 s windows starting at 5, 10, …, 25 s (a sequence of 8–10 s uses the window at 0 s). Kinetic features average velocity and acceleration over ± 2 frames, as in the Bailando code, and features are normalized by the real dance of the same sequences. Fréchet distances are taken against those sequences and averaged over datasets in Table III; Div_k is the mean pairwise distance of normalized kinetic windows.

f) Laban movement qualities: $\mathrm{FID}_{\mathrm{LMA}}$ is the Fréchet distance over the 55 Laban movement descriptors of Turab et al. [31], and Table VI also gives one distance per quality: Weight (kinetic energy), Time (acceleration), Flow (jerk), Space (paths, reach, height, travel and turning), Shape (body extent) and Body (coordination between limbs). Windows and normalization are those of FID_k. Every score separates real dance from itself with one second looped, far beyond the distance between two halves of its songs (on the 721 sequences, $\mathrm{FID}_{\mathrm{LMA}}$ 1443.7 against 15.2, and 11.7–555.5 against 1.1–3.1 for the six qualities). Commit Forcing lowers $\mathrm{FID}_{\mathrm{LMA}}$ ($-69.7\ [-76.1, -63.4]$ against Teacher Forcing). On the 42 songs, AHEADance is the closest to real dance in $\mathrm{FID}_{\mathrm{LMA}}$ (35.0 against at least 58.6), Space, Shape and Body, with every 95% CI excluding zero, and on Weight, Time and Flow it does not differ significantly from MEGADance, the closest there.

g) Baselines: The released generators were trained on AIST++ or FineDance. For Table II, each continues from its released weights on the AIOZ-GDANCE training songs of AHEADance (every dancer of every training sequence; 4,808 clips, 48.3 h) with its official training code, so it keeps its own corpus and adds ours; none of the 42 test songs is among them. Each stage passes over every training frame about as often as AHEADance's training does (about 55 times), in whole epochs nearest to update targets of 3,765 for EDGE, 3,621 per stage for Lodge, 18,777 and 12,517 for MEGADance's FSQ and GPT, 75,100 for each stage of Bailando++, and 300,424 and 75,100 for Danceba's VQ and GPT. Token generators train at the official learning rate, the other stages at one tenth of it. The strictly online rows use the same weights. With the released weights (Table VII), AHEADance's ChoreoSync offset accuracy is 28.6% against at most 16.7%, and its MMR 0.187 against at most 0.010.

Bailando++ and Danceba decode greedily, as released. Consecutive motion tokens of these training songs repeat often (26–31%), and after training on them the greedy decoders settle on a repeated token: 96% of Bailando++'s frames and 57% of Danceba's are still (joint speed below 5 deg/s; real dance 1%), which lowers their Div_k.

Table II reports no BAS. The common implementation [12] averages, over music beats, the distance to the nearest motion

TABLE V
CHUNK REPRESENTATION ON THE SAME 721 SEQUENCES

Setting	ChoreoSync↑	MMR↑	BAS↑	Tempo↑	Low act.↓	Span↑	FID_k ↓	FID_g ↓	Div_k
AHEADance (D+C)	24.2	**0.126**	0.262	56.8	**10.3**	**0.732**	**16.7**	**46.3**	2.94
Discrete only (D)	20.4	0.078	0.283	51.4	15.1	0.676	26.2	64.9	2.44
Continuous only (C)	**28.0**	0.054	0.270	**57.4**	34.6	0.589	28.5	81.3	2.60

Scores and sequences as in Table III. Bold: best in the column; BAS is not ranked.

TABLE VI
FID_{LMA} AND LABAN MOVEMENT QUALITIES

	Real dance		721 sequences					
Quality	Split	1 s loop	AHEADance	w/o future	TF	w/o closure	D	C
FID_{LMA}	15.2	1443.7	33.6	41.7^{+}	103.3^{+}	76.7^{+}	47.1^{+}	52.3^{+}

		Online, strict			Offline, released				
42 songs	AHEADance	Bailando++	Danceba	MEGADance	EDGE	Lodge	MEGADance	Bailando++	Danceba
FID_{LMA}	35.0	858.2^{+}	578.2^{+}	637.8^{+}	104.4^{+}	80.0^{+}	58.6^{+}	168.3^{+}	256.0^{+}
Weight (force)	8.0	122.2^{+}	62.9^{+}	66.8^{+}	7.8	7.9	3.8	31.7^{+}	47.2^{+}
Time (suddenness)	7.8	146.9^{+}	117.6^{+}	95.8^{+}	18.3^{+}	8.9	5.6	21.5^{+}	52.1^{+}
Flow (jerkiness)	7.2	433.4^{+}	254.8^{+}	365.3^{+}	26.8^{+}	9.7^{+}	4.7	34.1^{+}	58.3^{+}
Space (paths, reach)	5.0	106.2^{+}	95.7^{+}	83.5^{+}	27.0^{+}	24.5^{+}	20.3^{+}	41.6^{+}	48.5^{+}
Shape (body extent)	1.0	11.2^{+}	6.8^{+}	4.3^{+}	6.2^{+}	6.2^{+}	4.6^{+}	9.1^{+}	8.8^{+}
Body (coordination)	4.1	21.6^{+}	24.3^{+}	14.8^{+}	14.4^{+}	19.2^{+}	13.0^{+}	17.1^{+}	28.7^{+}

Fréchet distance to real dance (lower is closer). Split: between two halves of the real dance's songs. 1 s loop: real dance with one second looped. TF: Teacher Forcing; D, C: discrete-only and continuous-only. $^{+}/^{-}$: AHEADance closer / farther than this model (95% bootstrap CI of the difference over songs excludes zero).

TABLE VII
RELEASED WEIGHTS ON THE SAME 42 SONGS

Method	Trained on	FID_k ↓	Div_k	Sync↑	MMR↑	BAS↑	Tempo↑
Real dance	–	–	6.22	69.0	0.245	0.247	80.2
Online (no future music)							
AHEADance	6 datasets	18.8	2.22	**28.6**	**0.187**	0.266	**57.4**
Bailando++, strict	AIST++	153.5	14.77	11.9	0.002	0.250	56.6
Danceba, strict	AIST++	134.8	16.51	11.9	−0.006	0.261	50.9
MEGADance, strict	FineDance	79.1	8.44	11.1	0.003	0.251	48.2
Offline, released							
EDGE	AIST++	24.7	5.97	16.7	0.010	0.260	55.2
Lodge	FineDance	21.1	4.32	14.3	0.008	0.272	49.6
MEGADance	FineDance	**11.7**	6.42	12.7	0.005	0.238	49.5
Bailando++	AIST++	148.1	15.66	14.3	0.001	0.244	54.8
Danceba	AIST++	115.2	16.25	2.4	−0.009	0.290	47.4

Songs, scores and AHEADance as in Table II, with each baseline's released weights; AHEADance's training data include other songs of this corpus. Online, strict: the released weights run on causal input, without retraining. Bold: best generator in the column; BAS is not ranked.

beat, so a dense train of motion beats scores high without any dance: beats every 0.2 s and no motion score 0.848 on these songs. Near-still motion jitters: the Bailando++ and Danceba rows of Table II have 3.2–11.1 motion beats per second against 1.0 for real dance, and BAS 0.40–0.80 against 0.247. Averaged over motion beats, as originally defined [1], every row and every fixed beat train scores 0.48–0.51 on these songs.

Real dance and AHEADance use the retargeting of OMG-Data; the baselines' human motion is retargeted by a re-seeded GMR route [39]. On 39 FineDance songs with real dance through both routes, FID_k between the routes is 2.8–7.5, FID_{LMA} 6.4 and the MMR difference -0.013 $[-0.046, 0.016]$; FID_g between them reaches 126 and measures the route, so Table II leaves it out. ChoreoSync's estimates on the two routes agree within 0.2 s for 23 of 39 real dances.

The strictly online variants keep each model's start tokens and greedy decoding. Their music features are recomputed at every generation step from at most the last 10 s of heard audio, with unheard frames at the model's padding value, and their non-causal token decoders run on the 32 most recent tokens; streaming them without look-ahead leaves seams at token boundaries (for Bailando++, the mean absolute second difference of the pose rises from 0.0035 to 0.0305), and each token of look-ahead would add 0.133 s of delay. In the music-stop test, none of the strictly online variants reacts before the change (the earliest 0.20 s after it). Danceba outputs joint positions, which we fit with the neutral SMPL body (median joint error about 11 mm); both of its variants start from the first token pair of one AIST++ training dance, drawn per song. MEGADance and Lodge receive three drawn styles.

We trained DiscoForcing [6] with the official code, data and schedule and checked it against its reported FID_k with the official script before scoring any of its motion. Neither of two checkpoints (360k steps, and a 300k-step one chosen afterwards) reached it on both test sets within the $1.2\times$ tolerance fixed in advance, so we compare its interface (Table I) rather than its scores.

h) Style on unheard songs: Of the 30 FineDance songs of Fig. 4(b), which AHEADance never heard, we drop the four Mix songs and the three without real dance, leaving 23 (15

TABLE VIII
STYLE ON 23 FINEDANCE SONGS UNHEARD BY AHEADANCE

Method	Trained on	Music	Acc.	Follow↑	AHEADance−X
Real dance	–	–	100.0	47.4	–
Online (no future music)					
AHEADance	6 datasets	online	85.5	32.6	–
Bailando++, strict	AIST++	online	39.1	3.0	29.6 [10.1, 49.0]
Danceba, strict	AIST++	online	60.9	6.7	25.9 [8.7, 45.5]
MEGADance, strict^{s}	FineDance	online	60.9	−0.2	32.8 [12.8, 54.5]
Offline, released					
EDGE	AIST++	35 s	65.2	1.0	31.6 [12.8, 52.4]
Lodge^{s}	FineDance	68 s	73.9	29.6	3.0 [−6.9, 12.8]
MEGADance^{s}	FineDance	69.3 s	91.3	41.5	−8.9 [−18.8, −1.0]
Bailando++	AIST++	37.5 s	39.1	6.1	26.5 [6.7, 45.7]
Danceba	AIST++	37.5 s	73.9	19.0	13.6 [0.2, 29.2]

FineDance songs whose audio AHEADance never heard (15 Street, 8 Classic; Appendix A-h). Real dance and the released generators are retargeted by the re-seeded GMR route (Appendix A-g). Trained on and strict as in Table II; ^{s}: given each song's FineDance genre. Music: length of each song an offline generator receives; all motion is scored on its first 35.2 s. Acc.: style accuracy (%, descriptive). Follow: style following (points). AHEADance−X: paired difference [95% CI] over songs. Style following does not penalize labels absent from the song set, such as Folk here, so it can rank models differently from accuracy. Bailando++ and Danceba: one sample per song; others three. Real dance here is one dance per song from its start, on Classic and Street songs only; the 84.2% of Table IV covers 76 FineDance sequences of all three styles.

Street, 8 Classic; Table VIII). AHEADance dances each from the start with three seeds; the other generators follow their FineDance protocols, and all are scored on the first 35.2 s. Each song is scored by a classifier fitted without its fold or songs sharing its audio, on six 5 s windows from 5 s with a majority vote. AHEADance, EDGE, Lodge and offline MEGADance have every Street sample classified as Street, so their differences come from the eight Classic songs. On the FineDance clips of Table III, which are cut from test and validation sequences and often start mid-song, AHEADance's style following on the 21 songs it never heard, weighted by clip, is 5.5 $[-14.6, 27.8]$, not significant (real dance 46.0).

i) Whole songs: Over whole songs the movement does not decay: on ten long FineDance songs (108–243 s), low activity is 7.6% in the first minute and at most 2.8% in each later minute, and joint speed does not decline. In simulation SONIC executes all ten generated songs without a fall, with a body-link position error of 28 mm.

j) Execution in simulation: SONIC uses the same pre-trained controller for all methods. Every simulated trial counts, including those with falls or resets; it is scored over the complete generated clip and succeeds if there is no reset and the pelvis-height error stays at most 0.25 m throughout. Position error is mean Euclidean error over 29 pelvis-relative body links; joint error is the mean absolute joint-angle error.

k) Music stops and returns: EDGE, Lodge and MEGADance keep dancing through the silence at 1.03–1.77 rad/s although they receive the whole programme in advance. Each programme plays 20 s of a song, interrupts it with 60 s of silence and then resumes it. On the 16 interruption songs from the test split, the robot executed by SONIC in simulation responds after a median 1.03 s, and the continued-music control never reaches sustained low activity. Per event, the paired median difference between measured and generated times is at most 0.07 s in magnitude. Across all 28 programmes the robot then holds still: it does not move again during the 60 s silence and is at rest when the music returns. With the music continuing, false stops number 0 for AHEADance, EDGE and the strictly online Bailando++ and MEGADance, 2 for Lodge and the strictly online Danceba, 6 for Bailando++, 7 for MEGADance and 11 for Danceba.

The 28 programmes cover 27 songs: the 16 test-split songs, nine from the training split and two from validation; test-split song 211 is run twice. A response starts when the interrupted run is closer to the standing pose than the continued run by 0.05 rad for 0.5 s while moving toward it; near-standing then requires joint speed ≤ 0.3 rad/s and pose error ≤ 0.1 rad for 1 s. The standing pose is the SONIC standing pose, which is also the generator's initial physical boundary. Resumption is timed only for songs still at rest when the music returns. Generated references and SONIC-measured joints are scored on one content clock. Times exclude the delivery buffer, set to 1.6 s here for safety, so audio stop to measured near-standing takes 3.92 s on the wall clock. With the official low-latency SONIC (0.32 s preview) and a 0.8 s delay, AHEADance runs the eight programmes of Appendix A-l in a real-time MuJoCo loop: every reference is ready at least 0.31 s before its deadline, no delivery deadline is missed, the pelvis stays above 0.65 m, and joints track with a median error of 6.3°.

l) Physical G1: Twelve runs play six songs with 5 s of silence before and after 30 s of music (eight runs) and two 80 s programmes of three songs separated by 5 s silences (four runs), giving 20 music starts and 20 stops. Near-standing follows each music stop after 1.47–2.00 s, with no rebound, and sustained dancing follows each start after 0.37–1.23 s (medians in Sec. IV-E). Near-standing uses the thresholds above on measured joints, and sustained dancing requires joint speed above 0.3 rad/s and a joint range over 1 s above 0.1 rad on average or 0.2 rad at the 90th percentile, for 1 s; while the music plays, from the onset of sustained dancing to the next stop, the robot never meets the near-standing criterion (462 s of dancing; at most 0.43 s in a row). The measured delay from reference to execution is 1.27 s.

On a further day the music was played through the loudspeaker and captured by a microphone beside the robot, with the same model and delivery delay. The robot danced in eight runs, two of them on song 071, without a fall or safety stop (supplementary video).

m) Intervals: We use 95% paired bootstrap intervals (10,000 draws) over songs, so that sequences sharing a song are drawn together, averaging sampling seeds within a sequence before resampling. Fréchet-distance intervals resample songs within each dataset, recompute the set-level distance in each of 400 draws (300 for the 42 songs) and pair the two models on the same draw; Div_k has no interval. Intervals condition on the evaluated models, without multiplicity correction.

n) Timing: The times in Fig. 5 cover 1,200 chunks of the 12 physical runs; the slowest chunk is committed 177 ms after its audio cutoff, within the 0.34 s commit deadline.