WHICH MOVE, WHICH MOMENT? CUEDANCE: EFFICIENT RETRIEVAL-AUGMENTED MUSIC-TO-DANCE GENERATION VIA CHOREOGRAPHIC WORLD MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

People naturally move to music, while choreographic experience helps dancers determine which move fits which moment. This observation motivates us to view music-to-dance generation as two complementary capabilities: continuous responses to music and the selective use of motion experience at key moments. Existing retrieval-augmented methods introduce real motion references, but the temporal allocation of sparse references and sequential refinement dependent on an initial draft still limit their use of memory. We present CueDance, which connects what motion to retrieve with when to introduce it through a choreographic world model. Anticipate–Attribute–Anchor (3A) proposes motion hypotheses from musical transitions, selects reference moments based on explanatory gain (reduction in prediction error), and retrieves real motions as anchors. Music-driven retrieval runs in parallel with flow sampling, while Mid-Flow Anchor Inpainting integrates the references into the same generation pass. A part-aware generator and local-to-global flow supervision further support natural transitions around the references and efficient synthesis. CueDance achieves state-of-the-art dance generation performance on FineDance and AIST++, with a user study and systematic ablations further validating the effectiveness of selective choreographic memory.



Figure 1: CueDance, trained only on FineDance, generates full-song choreography for unseen music from *Only by Chasing the Wind*. Both PCG boxes belong to one sampling pass. CWM: Choreographic World Model. ***See the supplementary demo.***

1 INTRODUCTION

People naturally respond to music through movement: a drumbeat prompts a step, a rising melody prompts a stretch, and rhythmic accents and pauses change motion's speed, amplitude, or force. Music cognition research links these responses to beat prediction and sensorimotor coupling (Janata et al., 2012; Matthews et al., 2023). Beyond this immediate response, dance training further shapes the perception of learned movements (Calvo-Merino et al., 2005), and choreographic experience also helps dancers judge which movement suits the music and when to perform it. Motivated by these

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.

findings, we pair a continuous music-conditioned generator with a choreographic world model that draws on real motion memory to supply references at key moments.

Music-to-dance generation has advanced from pose prediction and discrete composition (Li et al., 2021; Siyao et al., 2022; 2023; Kim et al., 2022; Wang et al., 2024) to editable diffusion, hierarchical long-sequence generation, and efficient flows (Tseng et al., 2023; Li et al., 2023; 2024; 2025; Zhang et al., 2024; Yang et al., 2025a; 2026b; Ji et al., 2026). Genre experts, rhythm-aware gating, and few-step sampling improve quality, synchronization, and efficiency (Yang et al., 2025b; 2026a; Fan et al., 2025). Continuous latents also support compact representations and sequential generation (Xiao et al., 2025; Chen et al., 2023). Across these methods, choreographic knowledge resides in model parameters.

Retrieval-augmented methods instead consult real motion libraries at inference time. Music–motion matching retrieves reference segments or dances (Huang et al., 2025; Li et al., 2026), while motion planning specifies motion types, durations, and timing (Cai et al., 2026). For speech-driven gesture generation, a sparse-anchoring method queries a pose database with a generated clip's tail and refines the clip from a cached state (Zhang et al., 2026). However, neither content matching nor explicit planning measures how much a reference helps at a musical transition under a sparse budget, and draft-based retrieval also makes generation, retrieval, and refinement sequential. We therefore ask which musical moments most warrant a reference and how retrieval can overlap generation within one sampling pass.

Predictive world modeling offers a way to answer these questions. Joint-embedding predictive architectures (JEPA) learn predictable latent structure, while action-conditioned models further relate current and next states through actions (Assran et al., 2023; 2025; Tuncay et al., 2025). Because the music track is given in advance, both transition endpoints are available. We therefore propose a motion hypothesis from these states and measure how much it improves prediction of the next music state. This explanatory gain estimates the value of placing a reference there. The hypothesis thus indicates what to retrieve, the explanatory gain indicates when to intervene, and motion memory supplies the reference.

CueDance generates successive windows conditioned on music and preceding motion, using retrieved motions as anchors (Figure 1). Its Choreographic World Model (CWM) implements Anticipate–Attribute–Anchor (3A), which predicts hypotheses for valid transitions within each window in a single batch and retrieves motions at moments with high explanatory gain. Because retrieval uses music alone, it runs in parallel with early flow sampling. Mid-Flow Anchor Inpainting (MAI) incorporates each reference midway through sampling at the current noise level so the remaining updates generate surrounding transitions, combining memory access and motion synthesis within one sampling pass.

For the continuous response to music, the Part-Aware Choreographic Generator (PCG) models part-specific temporal dynamics and coordinates motion through shared music context and cross-part features. Local-to-Global Flow Supervision (L2G-Flow) further combines velocity matching, forward-kinematics (FK) joint-position loss, and per-state distribution supervision. Together, these components produce coherent motion around sparse references at a fixed sampling budget.

We make three contributions. First, we propose CWM-driven 3A, which selects reference content and timing, and MAI for incorporation within one sampling pass. Second, we design a part-aware generator with per-state adversarial supervision. Third, we evaluate quality and efficiency on FineDance and AIST++, with a user study and ablations of reference timing, content, and generator design.

2 Related Work

Generative modeling and flow supervision. FineNet coordinates full-body motion through body and hand experts followed by refinement (Li et al., 2023), while Duolando combines part-specific units, music, and partner information (Siyao et al., 2024). FlowerDance and MACE-Dance jointly generate whole sequences offline and non-autoregressively (Yang et al., 2026a;b), whereas MEGADance uses sliding-window autoregression (Yang et al., 2025b). HoleGest coordinates body parts through compressed full-body conditioning (Cheng & Huang, 2025). However, in streaming generation, each short window is conditioned on previous motion, so temporal and coordination modules are repeatedly evaluated. To reduce this recurring cost, our ForeGate uses gated temporal mixing and our Unison Hint supplies compact cross-part hints. Beyond architectural efficiency, supervision also

matters for few-step quality. Flow Matching (FM) and Rectified Flow learn velocity fields (Lipman et al., 2023; Liu et al., 2023), while Shortcut and MeanFlow use cross-step self-consistency and mean-velocity modeling (Frans et al., 2025; Geng et al., 2025). Adversarial flows add distribution supervision (Lin et al., 2026a;b), as does HoleGest for large-step motion denoising (Cheng & Huang, 2025). However, neither local accuracy nor single-transition distribution matching during training ensures that successive sampling updates produce states following their target distributions. L2G-Flow therefore supplements FM and FK-based joint-position loss (FK-pos) with per-state distribution supervision.

Retrieval augmentation and choreographic world modeling. Choreographic units encode internal motion memory (Siyao et al., 2022; 2023), whereas external retrieval matches real motions to semantic or musical cues (Huang et al., 2025; Li et al., 2026; Mughal et al., 2025; Zhang et al., 2023). MotionRAG-Diff (MotionRAG in the tables) organizes and refines retrieved segments through diffusion, InfiniteDance conditions generation on retrieved dances, and TANGO interpolates transitions through audio–motion retrieval and diffusion (Huang et al., 2025; Li et al., 2026; Liu et al., 2025). Atomic-movement methods plan motion types, durations, and timing (Cai et al., 2026). For speech-driven gestures, SemTalk integrates sparse semantic gestures with rhythmic motion using frame-level scores (Zhang et al., 2025), while StreamTalk queries a pose database with a generated clip's tail and refines the clip from a cached midpoint (Zhang et al., 2026). However, content similarity does not measure a reference's value at each moment, and draft-based retrieval must wait for generated motion. To address both limitations, we build CWM on JEPA and action-conditioned latent models (Assran et al., 2023; Bardes et al., 2024; Assran et al., 2025; Fei et al., 2023; Tuncay et al., 2025; Riou et al., 2024) to select references from music alone using explanatory gain, so retrieval can overlap sampling (Sections 3.4–3.5).

3 METHOD

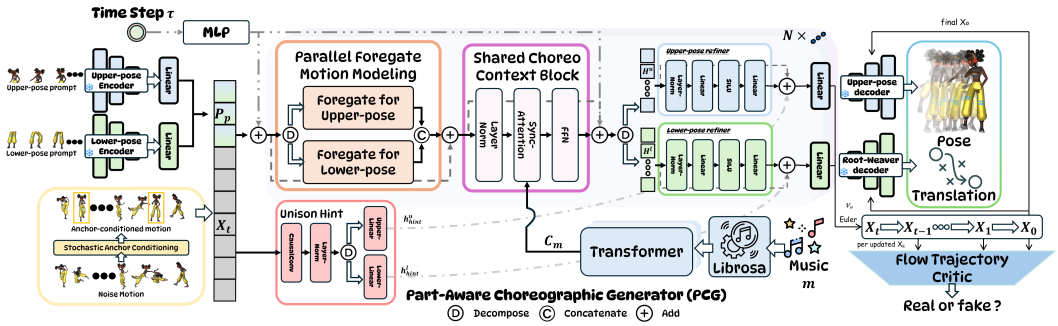


Figure 2: PCG and flow supervision. Frozen decoders and Root-Weaver recover poses and displacement. The shared Flow Trajectory Critic evaluates each post-update window separately, with adversarial losses averaged across steps during training.

3.1 OVERALL FRAMEWORK AND MOTION INTERFACE

Given music m, CueDance generates poses and global displacement window by window. Frozen causal codecs encode and decode upper- and lower-body poses as $X_0^b = E_b(P^b)$ and $\hat{P}^b = D_b(X_0^b)$, for $b \in \{u, l\}$. Their standardized latents form $X_0 = X_0^u \| X_0^l \in \mathbb{R}^{L \times D_a}$, with length L and D_a channels. Music is aligned with motion at compression ratio ρ for generation, CWM, and retrieval. From lower-body latents, a frozen Root-Weaver predicts local displacement increments, which are rotated using decoded pelvis orientation and accumulated into the global trajectory.

PCG predicts the velocity field over these latents, conditioned on music context C_m, full-window state X_t, and a Pose Prompt P_p from the previous window's ending (Figure 2). Sampling evolves $X_T = \epsilon$ to X_0, where $\tau = t/T$ and i indexes motion positions. Meanwhile, CWM retrieves a motion anchor $a^\star$ for MAI to incorporate into the current state.

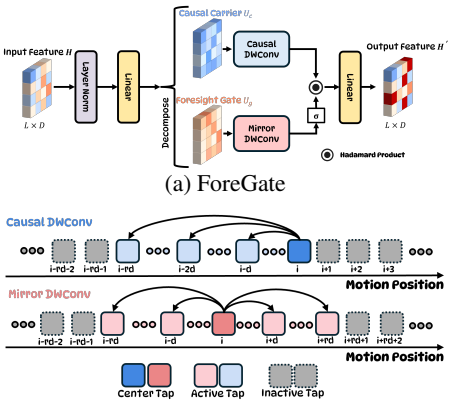
3.2 Part-Aware Choreographic Generator

To sustain continuous musical responses, each PCG unit applies ForeGate to upper- and lower-body features, followed by the Shared Choreo Context Block and part-specific refiners. The block's Sync-Attention attends from updated motion queries to keys and values projected from C_m. Unison Hint maps combined features through causal convolution, layer normalization (LN), and part-specific projections into cross-part hints for the refiners, which output v_Θ.

ForeGate. We project and split hidden features as $U_c \| U_g = \mathrm{Linear}_{\mathrm{in}}(\mathrm{LN}(H))$, then apply gated depthwise convolution (DWConv):

$$H' = \mathrm{Linear}_{\mathrm{out}}\big[\mathrm{DWConv}_c(U_c) \odot \sigma\big(\mathrm{DWConv}_g(U_g)\big)\big]. \tag{1}$$

Here, σ is sigmoid and $\odot$ is elementwise multiplication. The Causal Carrier reads past and current positions, while the Foresight Gate uses a two-sided receptive field to modulate updates (Figure 3). With fixed-width kernels, temporal mixing scales linearly with L.



(a) ForeGate

(b) Causal / Mirror DWConv

Figure 3: ForeGate: (a) causal content with mirrored gating. (b) past/current versus two-sided receptive fields of depthwise convolutions (DWConv).

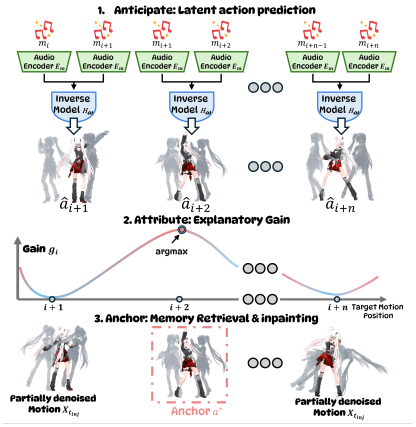


Figure 4: Our 3A retrieves anchors using music-driven hypotheses and explanatory gain. Each gain g_i is plotted at its target motion position $i+1$.

3.3 Local-to-Global Flow Supervision

L2G-Flow trains PCG with local flow-velocity and decoded joint-position objectives plus distribution matching of generated states. Given ground-truth (GT) latents X_0^{GT}, standard Gaussian noise ϵ, and $\tau \sim \mathcal{U}(0,1)$, we construct $X_t^{\mathrm{ref}} = (1-\tau)X_0^{\mathrm{GT}} + \tau\epsilon$. FM fits the latent flow velocity:

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}\big\| v_\Theta(X_t^{\mathrm{ref}}, \tau, c) - (\epsilon - X_0^{\mathrm{GT}}) \big\|_\Omega^2. \tag{2a}$$

Here, c contains music and available motion conditions. The set Ω marks generated latent positions and Ω_f their frames. FK-pos supervises decoded joint positions from the clean estimate $\hat{X}_0 = X_t^{\mathrm{ref}} - \tau v_\Theta(X_t^{\mathrm{ref}}, \tau, c)$:

$$\mathcal{L}_{\text{FK-pos}} = \mathbb{E}\big\| \Phi(\hat{X}_0) - \Phi(X_0^{\mathrm{GT}}) \big\|_{\Omega_f}^2. \tag{2b}$$

The map $\Phi = \mathrm{FK} \circ D_{\mathrm{pose}}$ computes root-relative joint positions from decoded poses, using given motion in known regions.

Per-state adversarial (ADV) supervision. FM learns at interpolated reference states, whereas sampling recursively evaluates model-generated states. To supervise these sampled states, we unroll T Euler updates from $X_T = \epsilon$:

$$X_{k-1} = \Pi_{k-1}\Big[X_k - \frac{1}{T} v_\Theta(X_k, k/T, c)\Big], \qquad k = T, \ldots, 1. \tag{3}$$

The projection Π_k maps known positions to the corresponding noise level. We match each generated post-update state $k = 0, \ldots, T-1$ to $X_k^{\mathrm{ref}} = (1-\tau_k)X_0^{\mathrm{GT}} + \tau_k\epsilon$, with $\tau_k = k/T$ and identical conditions, time grid, and known-position treatment.

The time-conditioned Flow Trajectory Critic $D_\eta(X_k, \tau_k, c)$, shared across steps, evaluates each complete post-update window separately. We train it with averaged per-state hinge losses and R_1 regularization, while the generator combines maximization of the mean critic score with FM and FK-pos:

$$\mathcal{L}_{\text{ADV}} = -\mathbb{E}\left[\frac{1}{T}\sum_{k=0}^{T-1} D_\eta(X_k, \tau_k, c)\right], \qquad \mathcal{L}_{\text{PCG}} = \mathcal{L}_{\text{FM}} + \lambda_{\text{FK}}\mathcal{L}_{\text{FK-pos}} + \lambda_{\text{ADV}}\mathcal{L}_{\text{ADV}}. \tag{4}$$

3.4 Choreographic World Model and 3A Retrieval

Action-conditioned music representations. To complement continuous generation, CWM selects reference motions and timing from music transitions. Each music segment m_i aligns with a latent position, with transition motion a_{i+1} supplied by the frozen pretrained pose encoder. We jointly train the student music encoder and forward model F_ψ using student states s_i^{stu} and stop-gradient targets $\overline{s_i}$ from an exponential moving average (EMA) teacher:

$$\mathcal{L}_{\text{wm}} = \mathbb{E}_i\Big[d\big(F_\psi(s_i^{\text{stu}}, a_{i+1}), \overline{s_{i+1}}\big) + d\big(F_\psi(s_i^{\text{stu}}, 0), \overline{s_i}\big)\Big]. \tag{5a}$$

Here, d is the mean squared error (MSE) over feature dimensions. The forward model learns to predict the next music state under motion conditioning and preserve the current state under zero-action conditioning. We then freeze the final teacher as E_m and fix F_ψ, before separately training an inverse model H_ω on $s_i = E_m(m_i)$ to recover standardized motion latents:

$$\mathcal{L}_{\text{inv}} = \mathbb{E}_i\, d\big(H_\omega(s_i, s_{i+1}), a_{i+1}\big). \tag{5b}$$

Anticipate. For all valid transitions within a window, H_ω predicts $\hat{a}_{i+1} = H_\omega(s_i, s_{i+1})$ in a single batch from music alone. These hypotheses use PCG's standardized coordinates and serve as retrieval queries (Figure 4).

Attribute. Explanatory gain measures how much a motion hypothesis reduces prediction error relative to the learned state-persistence baseline. Both errors are evaluated against the observed next music state:

$$g_i = d\big(F_\psi(s_i, 0), s_{i+1}\big) - d\big(F_\psi(s_i, \hat{a}_{i+1}), s_{i+1}\big). \tag{6}$$

We select $i^\star = \arg\max_{i\in\mathcal{I}} g_i$, where $\mathcal{I}$ contains transitions targeting positions to be generated and larger g_i indicates greater error reduction. We place one anchor at $i^\star + 1$, as this setting gives the lowest Fréchet inception distances (FIDs) and highest diversity among tested counts of 1, 4, and 8 (Appendix C.2).

Anchor. The fixed memory $\mathcal{M} = \{v_j\}_{j=1}^{N_{\text{mem}}}$ contains encoded training motions. Using the selected hypothesis as a query, we retrieve its nearest neighbor under cosine similarity:

$$j^\star = \arg\max_j\ \nu(\hat{a}_{i^\star+1})^\top \nu(v_j), \qquad a^\star = v_{j^\star}. \tag{7}$$

The map ν flattens and L_2-normalizes queries and keys for similarity, while $a^\star$ retains standardized latent scale for generation. Hypotheses and explanatory gains are computed in a single batch, but only the selected transition accesses memory.

3.5 Parallel Anchor Incorporation and Long-Sequence Generation

MAI incorporates retrieved anchors into PCG's ongoing sampling pass. Because 3A's queries and explanatory gains depend only on music, retrieval overlaps early flow updates. During ordinary PCG training, Stochastic Anchor Conditioning (SAC) exposes PCG to history only, anchors only, both, or neither, providing the inpainting interface that MAI reuses at inference.

Let M_A mark the selected position and A store the clean anchor there. At injection level t_{inj}, MAI forms $A_t = (1 - t/T)A + (t/T)\epsilon$ using the current trajectory's initial noise. After each Euler update and history projection, it applies:

$$X_t \leftarrow (1 - w_t M_A) \odot X_t + w_t M_A \odot A_t. \tag{8}$$

Here, $w_t = t/t_{\text{inj}}$ for $0 \le t \le t_{\text{inj}}$ and zero before incorporation. The anchor value and position also enter the motion-conditioning interface. The reference thus matches the current noise level as its weight decreases from 1 to 0, leaving the velocity field to determine the final motion. MAI reuses the original T evaluations and requires no separate training.

Each completed window's last L_p latents condition the next window as its Pose Prompt. We retain this history as conditioning, concatenate newly generated regions, then decode poses and accumulate root displacement. Local anchor refinements can therefore propagate to subsequent windows through generated history.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on FineDance (Li et al., 2023), with 7.7 hours across 16 categories at 30 frames per second (FPS), and AIST++ (Li et al., 2021), with 5.2 hours across 10 categories. Following Lodge and FlowerDance (Li et al., 2024; Yang et al., 2026a), we generate 1,024-frame (34.13 s) FineDance sequences. For AIST++, we retain 60 FPS and follow Danceba's protocol (Fan et al., 2025), computing FID and diversity (DIV) from the first 1,200 generated frames (20 s).

Table 1(a): Generation quality on FineDance. InfiniteDance is reevaluated; MotionRAG: stage2; MACE-Dance: Motion Expert.

Method	$\text{FID}_k \downarrow$	$\text{FID}_g \downarrow$	FSR $\leftrightarrow$	$\text{DIV}_k \uparrow$	$\text{DIV}_g \uparrow$	BAS $\uparrow$
Ground Truth	—	—	0.216	9.94	7.54	0.201
Bailando [CVPR'22]	82.81	28.17	0.188	7.74	6.25	0.202
EDGE [CVPR'23]	94.34	50.38	0.200	8.13	6.45	0.212
FineNet [ICCV'23]	65.15	23.81	0.192	5.84	5.19	0.219
Lodge [CVPR'24]	50.00	35.52	0.028	5.67	4.96	0.226
MEGADance [NeurIPS'25]	50.00	<u>13.02</u>	0.243	6.23	6.27	0.226
FlowerDance [ECCV'26]	29.73	19.59	0.147	8.42	<u>7.18</u>	0.232
MACE-Dance [SIGGRAPH'26]	<u>17.83</u>	25.09	<u>0.210</u>	**10.30**	**8.09**	0.229
DiscoForcing [ICML'26]	23.84	**8.62**	0.142	5.98	5.99	0.225
MotionRAG [arXiv'25]	32.25	57.63	—	<u>8.94</u>	3.75	**0.263**
InfiniteDance [ECCV'26]	24.46	29.93	0.236	7.37	5.44	0.225
CueDance (Ours)	**12.39**	16.17	**0.219**	<u>8.94</u>	6.71	<u>0.235</u>

Table 1(b): Generation quality on AIST++. MotionRAG: stage2; Danceba: Average.

Method	$\text{FID}_k \downarrow$	$\text{FID}_g \downarrow$	$\text{DIV}_k \uparrow$	$\text{DIV}_g \uparrow$	BAS $\uparrow$
Ground Truth	—	—	8.19	7.45	0.2374
FACT [ICCV'21]	35.35	22.11	5.94	6.18	0.2209
Bailando [CVPR'22]	28.16	<u>9.62</u>	7.83	6.34	0.2332
EDGE [CVPR'23]	42.16	22.12	3.96	4.61	0.2334
E3D2 [AAAI'24]	26.25	**8.94**	**7.96**	6.49	0.2232
Lodge [CVPR'24]	37.09	18.79	5.58	4.85	0.2423
MEGADance [NeurIPS'25]	25.89	12.62	5.84	6.23	0.2380
FlowerDance [ECCV'26]	20.50	15.75	6.95	6.52	0.2270
DiscoForcing [ICML'26]	18.87	11.57	6.76	6.31	0.2440
Danceba [ICCV'25]	<u>15.48</u>	13.57	<u>7.85</u>	**7.49**	**0.2779**
MotionRAG [arXiv'25]	26.23	17.66	5.62	3.79	<u>0.2545</u>
CueDance (Ours)	**11.79**	13.26	7.65	<u>6.84</u>	0.2390

Throughout, bold and underlining mark best and second-best results, excluding ground truth.

Metrics. FID-k/FID-g and DIV-k/DIV-g measure distributional discrepancy and motion diversity in kinetic and geometric feature spaces (Li et al., 2021; Siyao et al., 2022; Heusel et al., 2017). Beat alignment score (BAS) measures music–motion beat alignment (Li et al., 2021; Siyao et al., 2022; Li et al., 2024). We compute foot slide ratio (FSR) following Yang et al. (2026a), based on Tseng et al. (2023); Li et al. (2024). Since sliding can be choreographic and lower FSR may reflect restricted motion, we rank its absolute deviation from GT ($\leftrightarrow$) to assess consistency with ground-truth foot motion. FPS measures generation throughput.

Implementation details. Music frames use 35-dimensional features (McFee et al., 2015), and motion frames use 6D rotations of 24 Skinned Multi-Person Linear (SMPL) joints (Loper et al., 2015; Zhou et al., 2019). A frozen codec compresses 240 frames into 60 tokens of 254 dimensions. Subsequent windows retain 12 history tokens and generate 48 new tokens. Motion memory uses only the dataset's training split. Inference uses 20 Euler steps, incorporating one reference token after step 10. Ablations use FineDance; Appendices A–C give implementation, evaluation protocols, and supplementary analyses.

4.2 Generation Quality and Efficiency

Quantitative comparison. Combining continuous generation with selective motion memory, CueDance achieves the lowest FID-k on both datasets (Table 1), reducing it by 30.5% relative to MACE-Dance on FineDance and 23.8% relative to Danceba on AIST++. It also outperforms retrieval baselines on both FIDs. On FineDance, CueDance retains high diversity, while its FSR of 0.219 closely matches the ground-truth value of 0.216.

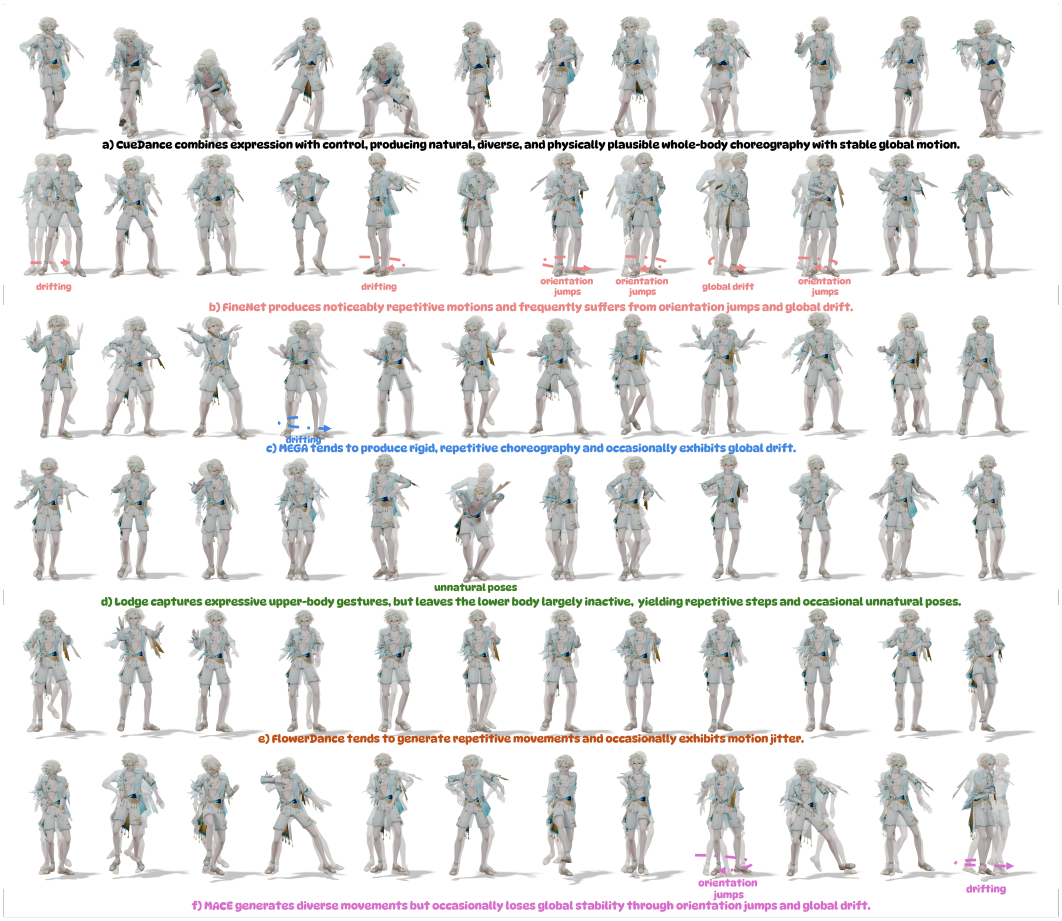


Figure 5: Comparison under the same music (MEGA: MEGADance; MACE: MACE-Dance). ***See the supplementary demo.***

Qualitative comparison. CueDance produces varied, coordinated motion with global stability (Figure 5). By contrast, MACE-Dance combines varied movements with orientation jumps and drift, while Lodge pairs expressive upper-body gestures with largely inactive legs and occasional unnatural poses. FineNet repeats globally unstable movements, MEGADance appears rigid and repetitive, and FlowerDance shows repetition and occasional jitter. These examples illustrate the complementary roles of ForeGate and Unison Hint: continuity within parts and coordination across them.

Generation efficiency. Because retrieval uses music alone, CueDance consults memory while sampling, achieving 994 FPS for 1,024 frames and 1,798 FPS for 10,240 frames (Table 2). It ranks second at both lengths, ahead of MACE-Dance's Motion Expert. The faster FlowerDance avoids inter-window dependencies by updating complete sequences jointly offline, whereas CueDance supports streaming generation and online retrieval.

User study. Thirty participants with dance experience rated six methods on 30 approximately 34-second in-the-wild music clips using a five-point scale. Six raters assessed each sequence with shared character, camera, and rendering settings, hidden method identities, and randomly ordered outputs. CueDance's distributional gains are mirrored by the highest dance quality (DQ), dance–

Table 2: Throughput (FPS), RTX 3090/Xeon Gold 5218. 1K/10K: 1,024/10,240 frames.

Method	FPS 1K ↑	FPS 10K ↑
Bailando (Siyao et al., 2022)	461	673
EDGE (Tseng et al., 2023)	372	589
FineNet (Li et al., 2023)	656	1137
Lodge (Li et al., 2024)	434	884
MEGADance (Yang et al., 2025b)	288	340
FlowerDance (Yang et al., 2026a)	**2008**	**4145**
MACE-Dance (Yang et al., 2026b)	812	1668
InfiniteDance (Li et al., 2026)	179	181
CueDance	<u>994</u>	<u>1798</u>

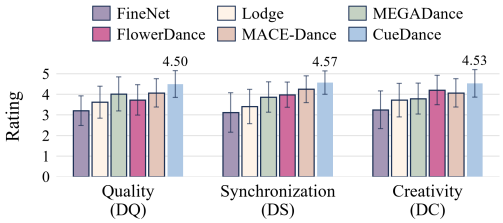


Figure 6: User ratings of quality (DQ), music synchronization (DS), and creativity (DC).

music synchronization (DS), and dance creativity (DC) ratings of 4.50, 4.57, and 4.53, respectively (Figure 6).

4.3 Selective Choreographic Enhancement with 3A

Reference timing. Does reference timing benefit more from musical salience or explanatory gain? Table 3 fixes the generator, motion-hypothesis prediction, memory, MAI, and one-anchor budget. Beat/Onset selects the strongest onset, while ΔMusic selects the largest consecutive CWM music feature change. Raw Error selects the largest mismatch between CWM's learned state-persistence prediction and the next music state. Subtracting the motion-conditioned residual lowers both FIDs relative to all three baselines, reducing FID-k by 28.5% over ΔMusic and 15.3% over Raw Error. Higher BAS and similar diversity to Raw Error further favor explanatory gain as a timing rule under the same anchor budget.

Table 3: 3A inference comparison with a shared SAC checkpoint and paired initial noise per run. Full 3A combines parallel retrieval with MAI. Nearest-neighbor retrieval (NNR) uses 20+10 evaluations; others use 20. —: unmeasured speed.

3A setting	FID_k ↓	FID_g ↓	DIV_k ↑	DIV_g ↑	BAS ↑	FPS ↑
When: Reference timing						
Beat/Onset	15.74	16.94	8.53	<u>6.78</u>	0.232	—
ΔMusic	17.34	17.94	8.41	6.52	0.224	—
Raw Error	14.62	16.83	8.89	6.73	0.229	—
What: Reference content						
SAC w/o anchor	20.78	18.29	**9.42**	**6.84**	0.221	—
Random Action	22.07	20.38	<u>9.32</u>	6.77	0.215	—
Direct Prediction	15.40	17.60	8.57	6.34	0.231	—
Retrieval and execution						
NNR	17.99	18.25	7.90	5.92	0.217	642
Serial 3A	<u>12.40</u>	<u>16.18</u>	8.96	6.67	**0.236**	<u>874</u>
3A	**12.39**	**16.17**	8.94	6.71	<u>0.235</u>	**994**

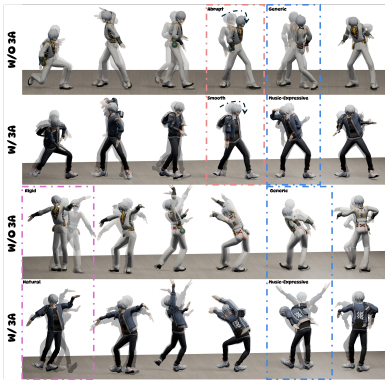


Figure 7: Our 3A preserves the main pattern while refining key poses and nearby transitions. Each row pair shares a SAC checkpoint, with anchors off/on (top/bottom). Red, purple, and blue boxes mark transitions, pose naturalness, and key-movement expression. ***See the supplementary demo.***

Reference content. Useful anchoring depends on matching content to the motion hypothesis. With timing and MAI fixed, retrieval lowers both FIDs relative to Direct Prediction, which inserts inverse-model outputs. However, Random Action raises FID-k above no anchors (22.07 versus 20.78). Retrieval lowers FID-k from 20.78 to 12.39 while DIV-k decreases modestly from 9.42 to 8.94. The hypothesis therefore guides retrieval, while retrieved motion provides the anchor.

Retrieval and execution. Following StreamTalk, NNR retrieves in raw action space using a generated draft as the query and L1 distance between FK joint coordinates, then refines the draft. Music-only retrieval lets 3A search in latent space by cosine similarity while sampling. Even with 30 evaluations, NNR has worse FIDs and BAS than 3A's 20-step pass. With other settings fixed, parallelizing Serial 3A raises throughput from 874 to 994 FPS (+13.7%) with nearly unchanged quality.

Local changes, sustained quality. With 3A, key poses and nearby transitions improve while the main movement pattern is preserved (Figure 7). To examine local correction and propagation, we anchor one window (Appendix C.8). The last complete window has almost no subsequent frames to influence, so anchoring it yields an essentially local FID-k reduction from 20.78 without anchors to 19.61. Anchoring only the first reaches 17.73, with Pose Prompts carrying changes forward, while anchoring every window reaches 12.39. Repeated anchoring reduces the decline in geometric and kinetic retention over time (Figure 8).

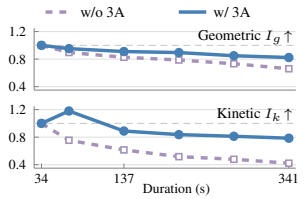


Figure 8: Relative quality retention (Appendix C.6).

Table 4(a): Temporal mixer ablation.

Temporal mixer	FID_k ↓	FID_g ↓	FSR ↔	DIV_k ↑	DIV_g ↑	BAS ↑	FPS ↑
Causal Trans.	17.76	20.05	0.182	8.69	5.92	0.214	855
BiMamba	17.03	19.93	0.265	6.85	5.99	0.232	192
Mamba	**11.67**	28.44	0.388	**9.33**	6.12	0.224	321
ForeGate	12.39	**16.17**	**0.219**	8.94	**6.71**	**0.235**	**994**

Table 4(b): Body coordination ablation.

Body coord.	FID_k ↓	FID_g ↓	FSR ↔	DIV_k ↑	DIV_g ↑	BAS ↑	FPS ↑
Joint attention	36.67	22.23	0.250	**10.25**	5.74	0.221	848.84
Cross attention	29.93	23.44	0.231	9.25	6.50	0.226	831.44
Unison Hint	**12.39**	**16.17**	**0.219**	8.94	**6.71**	**0.235**	**994.00**

Table 4(c): Supervision ablation.

Objective	FID_k ↓	FID_g ↓	FSR ↔	DIV_k ↑	DIV_g ↑	BAS ↑
FM	35.48	45.14	0.270	7.58	**8.34**	0.230
FM+ADV	18.36	21.95	0.207	9.24	6.51	0.221
FM+FK-pos	29.05	27.00	0.265	**9.98**	6.45	0.214
FM+FK-pos+ Endpoint	16.90	19.30	0.233	9.10	6.45	0.229
L2G-Flow	**12.39**	**16.17**	**0.219**	8.94	6.71	**0.235**

Table 4(d): Generative-method comparison.

Method	FID_k ↓	FID_g ↓	FSR ↔	DIV_k ↑	DIV_g ↑	BAS ↑
Rectified Flow (3× reflow)	26.34	21.09	0.222	**10.30**	6.24	0.218
Diffusion	37.30	27.81	0.270	7.32	6.51	0.226
MeanFlow	32.50	25.64	0.294	9.72	5.95	0.229
L2G-Flow	**12.39**	**16.17**	**0.219**	8.94	**6.71**	**0.235**

4.4 Generation Architecture and Supervision

ForeGate and Unison Hint. Table 4(a) changes only the temporal mixer. Although Mamba attains slightly lower FID-k, ForeGate gives better FID-g and foot-motion statistics. Compared with BiMamba, ForeGate lowers both FIDs while raising throughput from 192 to 994 FPS. Unison Hint likewise improves quality and speed. With shared music context fixed and only coordination varying, it outperforms both attention alternatives in FIDs, BAS, and throughput (Table 4(b)).

Local-to-global supervision. FK-pos and ADV complement velocity matching (Table 4(c)): FK-pos constrains decoded joint positions, while ADV supervises distributions of generated states. Adding either to FM lowers both FIDs, and all three yield the lowest values. With FM and FK-pos fixed, replacing Endpoint's final-state ADV with per-state ADV lowers FID-k/FID-g by 26.7%/16.2%. Although DIV-k decreases slightly, DIV-g and BAS improve, supporting distribution constraints throughout sampling.

Generative-method comparison. Table 4(d) fixes PCG, FK-pos, and 20 sampling steps, with denoising diffusion implicit model (DDIM) sampling for Diffusion. L2G-Flow lowers FIDs and raises BAS over all three alternatives. Relative to Rectified Flow with three reflow rounds, the strongest FID baseline, FID-k/FID-g fall by 53.0%/23.3%, with higher DIV-g but lower DIV-k. Together with the Endpoint ablation, these results favor supplementing local velocity and pose objectives with distribution constraints at each sampling step.

5 Conclusion

Inspired by dancers' use of movement experience, CueDance combines continuous generation with selective motion retrieval in one sampling pass. Evaluations on two datasets, a user study, and ablations demonstrate gains in quality and efficiency. Sparse anchors refine key poses and nearby transitions while preserving the main movement pattern, and propagation through Pose Prompts together with repeated anchoring extends these benefits across windows.

AI USE STATEMENT

We used generative AI tools primarily to polish the writing and improve the clarity, grammar, and readability of the manuscript. All AI-assisted revisions were reviewed and edited by the authors to ensure that they accurately reflected the intended technical meaning. The authors take full responsibility for the final content of this work, including all text, claims, and results.

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

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-Supervised Learning From Images With a Joint-Embedding Predictive Architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15619–15629, 2023. URL https://doi.org/10.1109/CVPR52729.2023.01499.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-Supervised Video Models Enable Understanding, Prediction and Planning. arXiv preprint arXiv:2506.09985, 2025. URL https://arxiv.org/abs/2506.09985.

Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. Revisiting Feature Prediction for Learning Visual Representations from Video. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=QaCCuDfBk2.

Xinhao Cai, Yixuan Sun, Minghang Zheng, Qingchao Chen, Xin Jin, Song-chun Zhu, and Yang Liu. Music-to-Dance Generation via Atomic Movements. arXiv preprint arXiv:2607.13978, 2026. URL https://arxiv.org/abs/2607.13978.

Beatriz Calvo-Merino, Daniel E. Glaser, Julie Grèzes, Richard E. Passingham, and Patrick Haggard. Action Observation and Acquired Motor Skills: An fMRI Study with Expert Dancers. *Cerebral Cortex*, 15(8):1243–1249, 2005. doi: 10.1093/cercor/bhi007. URL https://discovery.ucl.ac.uk/id/eprint/3041/.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18000–18010, 2023. URL https://doi.org/10.1109/CVPR52729.2023.01726.

Yongkang Cheng and Shaoli Huang. HoleGest: Decoupled Diffusion and Motion Priors for Generating Holisticly Expressive Co-Speech Gestures. In *International Conference on 3D Vision*, 2025. URL https://doi.org/10.1109/3DV66043.2025.00074.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language Modeling with Gated Convolutional Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 933–941. PMLR, 2017. URL https://proceedings.mlr.press/v70/dauphin17a.html.

Congyi Fan, Jian Guan, Xuanjia Zhao, Dongli Xu, Youtian Lin, Tong Ye, Pengming Feng, and Haiwei Pan. 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. URL https://arxiv.org/abs/2503.17340.

Zhengcong Fei, Mingyuan Fan, and Junshi Huang. A-JEPA: Joint-Embedding Predictive Architecture Can Listen. arXiv preprint arXiv:2311.15830, 2023. URL https://arxiv.org/abs/2311.15830.

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One Step Diffusion via Shortcut Models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/559a0998fab1d19b80e7e43a5852401c-Abstract-Conference.html.

Zhengyang Geng, Mingyang Deng, Xingjian Bai, Zico Kolter, and Kaiming He. Mean Flows for One-step Generative Modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2534. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/6d13e085b79d454da5910e4ca82a3d9d-Abstract-Conference.html.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Mingyang Huang, Peng Zhang, and Bang Zhang. MotionRAG-Diff: A Retrieval-Augmented Diffusion Framework for Long-Term Music-to-Dance Generation. arXiv preprint arXiv:2506.02661, 2025. URL https://arxiv.org/abs/2506.02661.

Petr Janata, Stefan T. Tomic, and Jason M. Haberman. Sensorimotor Coupling in Music and the Psychology of the Groove. *Journal of Experimental Psychology: General*, 141(1):54–75, 2012. doi: 10.1037/a0024208. URL https://doi.org/10.1037/a0024208.

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. In *International Conference on Machine Learning*, 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*, pp. 3490–3500, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Kim_A_Brand_New_Dance_Partner_Music-Conditioned_Pluralistic_Dancing_Controlled_by_CVPR_2022_paper.html.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2212.03741.

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*, pp. 1524–1534, 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*, 2025. doi: 10.1109/TPAMI.2025.3618016. URL https://doi.org/10.1109/TPAMI.2025.3618016. Early Access.

Ronghui Li, Zhongyuan Hu, Li Siyao, Youliang Zhang, Haozhe Xie, Mingyuan Zhang, Jie Guo, Xiu Li, and Ziwei Liu. InfiniteDance: Scalable 3D Dance Generation Towards in-the-wild Generalization. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2603.13375.

Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. AI Choreographer: Music Conditioned 3D Dance Generation With AIST++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13401–13412, 2021. URL https://arxiv.org/abs/2101.08779.

Jae Hyun Lim and Jong Chul Ye. Geometric GAN, 2017. URL https://arxiv.org/abs/1705.02894.

Shanchuan Lin, Ceyuan Yang, Zhijie Lin, Hao Chen, and Haoqi Fan. Adversarial Flow Models. In *International Conference on Machine Learning*, 2026a. URL https://arxiv.org/abs/2511.22475.

Shanchuan Lin, Ceyuan Yang, Zhijie Lin, Hao Chen, and Haoqi Fan. Continuous Adversarial Flow Models. In *European Conference on Computer Vision*, 2026b. URL https://arxiv.org/abs/2604.11521.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *International Conference on Learning Representations*, 2023. URL https://iclr.cc/virtual/2023/poster/11309.

Haiyang Liu, Xingchao Yang, Tomoya Akiyama, Yuantian Huang, Qiaoge Li, Shigeru Kuriyama, and Takafumi Taketomi. TANGO: Co-Speech Gesture Video Reenactment with Hierarchical Audio Motion Embedding and Diffusion Interpolation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/81a0a519bf7ef7757ff46df2065e7051-Paper-Conference.pdf.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2209.03003.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A Skinned Multi-Person Linear Model. *ACM Transactions on Graphics*, 34(6):248:1–248:16, 2015. URL https://doi.org/10.1145/2816795.2818013.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101v3.

Tomas E. Matthews, Jan Stupacher, and Peter Vuust. The Pleasurable Urge to Move to Music Through the Lens of Learning Progress. *Journal of Cognition*, 6(1):55, 2023. doi: 10.5334/joc.320. URL https://journalofcognition.org/articles/10.5334/joc.320.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and Music Signal Analysis in Python. In *Proceedings of the 14th Python in Science Conference*, 2015. URL https://brianmcfee.net/papers/scipy2015_librosa.pdf.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which Training Methods for GANs do actually Converge? In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3481–3490. PMLR, 2018. URL https://proceedings.mlr.press/v80/mescheder18a.html.

M. Hamza Mughal, Rishabh Dabral, Merel C.J. Scholman, Vera Demberg, and Christian Theobalt. Retrieving Semantics from the Deep: an RAG Solution for Gesture Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16578–16588, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Mughal_Retrieving_Semantics_from_the_Deep_an_RAG_Solution_for_Gesture_CVPR_2025_paper.html.

Alain Riou, Stefan Lattner, Gaëtan Hadjeres, Michael Anslow, and Geoffroy Peeters. Stem-JEPA: A Joint-Embedding Predictive Architecture for Musical Stem Compatibility Estimation. In *Proceedings of the 25th International Society for Music Information Retrieval Conference*, 2024. URL https://arxiv.org/abs/2408.02514.

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. URL https://arxiv.org/abs/2203.13055.

Li Siyao, Weijiang Yu, Tianpei Gu, Chunze Lin, Quan Wang, Chen Qian, Chen Change Loy, and Ziwei Liu. Bailando++: 3D Dance GPT With Choreographic Memory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14192–14207, 2023. doi: 10.1109/TPAMI.2023.3319435. URL https://doi.org/10.1109/TPAMI.2023.3319435.

Li Siyao, Tianpei Gu, Zhitao Yang, Zhengyu Lin, Ziwei Liu, Henghui Ding, Lei Yang, and Chen Change Loy. Duolando: Follower GPT with Off-Policy Reinforcement Learning for Dance Accompaniment. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=GW4j4n2cjH.

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*, pp. 448–458, 2023. URL https://arxiv.org/abs/2211.10658.

Ludovic Tuncay, Étienne Labbé, Emmanouil Benetos, and Thomas Pellegrini. Audio-JEPA: Joint-Embedding Predictive Architecture for Audio Representation Learning. In *IEEE International Conference on Multimedia and Expo*, 2025. URL https://arxiv.org/abs/2507.02915.

Zilin Wang, Haolin Zhuang, Lu Li, Yinmin Zhang, Junjie Zhong, Jun Chen, Yu Yang, Boshi Tang, and Zhiyong Wu. Explore 3D Dance Generation via Reward Model from Automatically-Ranked Demonstrations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 301–309, 2024. doi: 10.1609/aaai.v38i1.27783. URL https://ojs.aaai.org/index.php/AAAI/article/view/27783.

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*, 2025. URL https://arxiv.org/abs/2503.15451.

Kaixing Yang, Xulong Tang, Yuxuan Hu, Jiahao Yang, Hongyan Liu, Qinnan Zhang, Jun He, and Zhaoxin Fan. MatchDance: Collaborative Mamba-Transformer Architecture Matching for High-Quality 3D Dance Synthesis. arXiv preprint arXiv:2505.14222v1, 2025a. URL https://arxiv.org/abs/2505.14222v1.

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*, volume 38, 2025b. doi: 10.52202/085713-0366. URL https://papers.nips.cc/paper_files/paper/2025/hash/102bd3c1076dfe98043b19340ac44e08-Abstract-Conference.html.

Kaixing Yang, Xulong Tang, Ziqiao Peng, Xiangyue Zhang, Chubin Chen, Xukun Zhou, Puwei Wang, Hongyan Liu, and Jun He. FlowerDance: MeanFlow for Efficient and Refined 3D Dance Generation. In *European Conference on Computer Vision*, 2026a. URL https://media.eventhosts.cc/Conferences/ECCV2026/pdfs/5049.pdf.

Kaixing Yang, Jiashu Zhu, Xulong Tang, Ziqiao Peng, Xiangyue Zhang, Puwei Wang, Jiahong Wu, Xiangxiang Chu, Hongyan Liu, and Jun He. MACE-Dance: Motion-Appearance Cascaded Experts for Music-Driven Dance Video Generation. In *Proceedings of the ACM SIGGRAPH Conference*, 2026b. URL https://arxiv.org/abs/2512.18181.

Canyu Zhang, Youbao Tang, Ning Zhang, Ruei-Sung Lin, Mei Han, Jing Xiao, and Song Wang. Bidirectional Autoregressive Diffusion Model for Dance Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 687–696, 2024. URL https://arxiv.org/abs/2402.04356.

Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. ReMoDiffuse: Retrieval-Augmented Motion Diffusion Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 364–373, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zhang_ReMoDiffuse_Retrieval-Augmented_Motion_Diffusion_Model_ICCV_2023_paper.html.

Xiangyue Zhang, Jianfang Li, Jiaxu Zhang, Ziqiang Dang, Jianqiang Ren, Liefeng Bo, and Zhigang Tu. SemTalk: Holistic Co-speech Motion Generation with Frame-level Semantic Emphasis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13761–13771, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zhang_SemTalk_Holistic_Co-speech_Motion_Generation_with_Frame-level_Semantic_Emphasis_ICCV_2025_paper.html.

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://arxiv.org/abs/2608.01643.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the Continuity of Rotation Representations in Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5745–5753, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Zhou_On_the_Continuity_of_Rotation_Representations_in_Neural_Networks_CVPR_2019_paper.html.

A Implementation Details

A.1 Computing Environment

Training and routine inference. All model training, routine inference, and quality evaluations in this work were performed on the same workstation. The workstation has one NVIDIA GeForce RTX 5090 (32 GB), an Intel Core Ultra 9 285K CPU, and 62 GB of RAM. Its software environment consists of Ubuntu 20.04.6 LTS, NVIDIA driver 570.172.08, CUDA 12.8, Python 3.10.20, and PyTorch 2.10.0+cu128. All training stages use a batch size of 128 and 32-bit floating-point (FP32) precision.

FPS measurement. All FPS results in this work, including the efficiency comparison and all ablations, use the same benchmarking hardware as FlowerDance (Yang et al., 2026a): one NVIDIA RTX 3090 GPU and an Intel Xeon Gold 5218 CPU (2.30 GHz). In our measurements, we generate one sequence at a time (batch size 1). Section B.3 specifies the output lengths and the operations included in the timing.

A.2 Data and Normalization

For FineDance, we follow the data split and evaluation settings of Lodge (Li et al., 2024) and FlowerDance (Yang et al., 2026a), with 183 training source sequences and 19 final-evaluation source sequences. At 30 FPS, training windows contain 240 frames with a sliding stride of 10 frames. Evaluation uses 1,024-frame outputs.

For AIST++, we follow the data split and evaluation settings of Danceba (Fan et al., 2025), with 951 training source sequences and 40 final-evaluation source sequences. At 60 FPS, training windows contain 240 frames with a sliding stride of 20 frames. FID and DIV are evaluated on the first 1,200 generated frames.

In addition to the joint rotations described in the main text, each frame-level motion record contains 4 contact dimensions and 3 root-position dimensions, for a total of 151 dimensions. The music features comprise onset strength, 20-dimensional mel-frequency cepstral coefficients (MFCCs), 12-dimensional chroma energy normalized statistics (CENS), an onset indicator, and a beat indicator (McFee et al., 2015).

We standardize the motion latents channel by channel, computing the statistics from each channel's population moments. Let $z_{n,c}$ denote the encoded values and N_{stat} the number of positions used to

compute the statistics:

$$\mu_c = \frac{1}{N_{\text{stat}}} \sum_n z_{n,c}, \qquad \sigma_c = \sqrt{\max\Big(0, \frac{1}{N_{\text{stat}}} \sum\nolimits_n z_{n,c}^2 - \mu_c^2\Big)}. \tag{A.1}$$

$$X_{0,i,c} = \frac{z_{i,c} - \mu_c}{\sigma_c + 10^{-6}}, \qquad z_{i,c} = X_{0,i,c}(\sigma_c + 10^{-6}) + \mu_c. \tag{A.2}$$

We apply the inverse transform in Eq. (A.2) before decoding.

A.3 Motion Codec and Root-Weaver

The lower-body latent consists of 14-dimensional representations of nine joints, augmented with a linear embedding of local rotations over four frames. Table A.1 gives the architecture of each branch.

Table A.1: Network architectures of the pose codec and Root-Weaver.

Module	**Input**	**Latent**	**Output**
Upper-body pose	240×90	60×128	240×90
Lower-body pose	240×57	60×126	240×57
Root-Weaver	60×126	—	240×3

Dimensions are time $\times$ channels; — denotes no separate encoded latent.

Component	**Building blocks and role**
Upper-body encoder	Width 512; 2 stages of stride-2 causal convolutions.
Lower-body encoder	Width 512; nine-joint bottleneck. Input: 3 zeroed displacement placeholder channels.
Pose decoding	Upper-/lower-body latents $\rightarrow$ corresponding joint rotations. 2 stages: nearest-neighbor upsampling ($\times 2$) and causal convolutions.
Root-Weaver	$126 \rightarrow 512$ projection; causal residual blocks; 2 upsampling stages. Output: local displacement increments.

The last 54 dimensions of the lower-body branch's decoded output recover the rotations of nine joints. The first three dimensions are placeholder channels with zero-valued reconstruction targets and do not contribute to the final recovery of root translation.

Pose codec. During pretraining, we jointly use pose reconstruction, first- and second-order temporal differences, and a foot-displacement constraint under the ground-truth contact mask:

$$\mathcal{L}_{\text{pose-codec}} = \mathcal{L}_{\text{pose}} + \mathcal{L}_{\Delta} + \mathcal{L}_{\Delta^2} + 0.167\mathcal{L}_{\text{foot}}. \tag{A.3}$$

The pose and temporal-difference terms all use mean L1 errors, computed separately for the upper- and lower-body branches and then summed. The foot term compares FK foot-displacement vectors under the ground-truth support mask. Contact labels serve only as supervision masks for this loss, rather than prediction channels for the pose encoder. We freeze the codec after pretraining.

Root-Weaver. We train this causal decoder separately to predict a three-dimensional local displacement increment for each frame from the lower-body latent. Table A.1 gives its architecture. After reversing the standardization of each increment, we rotate it using the pelvis orientation decoded for the preceding frame and integrate the increments to obtain root positions:

$$\widehat{\Delta r_f} = R(\hat{\eta}_{f-1})\hat{\delta}_f^{\text{loc}}, \qquad \hat{r}_f = \hat{r}_0 + \sum_{i=1}^{f} \widehat{\Delta r_i}. \tag{A.4}$$

We set the first frame's displacement increment to zero and use a single root-position origin for the entire sequence. During training, the frozen encoder encodes ground-truth lower-body poses, and

the frozen pose decoder provides the orientations. During generation, the same modules receive generated latents. Let $e_f = \widehat{\Delta r_f} - (r_f - r_{f-1})$ denote the error in the world-space displacement increment. The training objective is:

$$\mathcal{L}_{\text{root}} = \text{mean}_f \|e_f\|_2 + \text{mean}_f \|e_f - e_{f-1}\|_2. \tag{A.5}$$

Both terms use vector L2 norms rather than mean squared error (MSE). The codec parameters remain frozen in subsequent stages, while PCG's FK-pos loss continues to backpropagate to the generated latents through differentiable decoding.

A.4 Network Configurations and Training Procedure

Table A.2: Network configurations of PCG, the Flow Trajectory Critic, and CWM.

Generator and critic

Component	Width / channels	Structure
PCG body branches	Hidden width 352	8 layers per upper/lower branch
ForeGate	64 per content/gating path	Radius 8; 9 radial weights per path
Flow Trajectory Critic	Hidden width 256	6 residual blocks

Transformer music encoders

Module	Projection	Layers	Heads	FFN width
PCG	$140 \to 512$	2	8	2048
CWM	$140 \to 256$	4	4	1024

CWM prediction MLPs

Model	Input width	Hidden widths	Output width
Forward	510	$512 \to 512$	256
Inverse	512	$512 \to 512$	254

ForeGate implementation. ForeGate uses multiplicative convolutional gating (Dauphin et al., 2017). Given dilation d and radius r, the content path reads $\{i - jd\}_{j=0}^{r}$, and the gating path reads $\{i + jd\}_{j=-r}^{r}$. The paths independently learn channel-wise radial weights and normalize them with softmax. The mirrored path divides each noncentral weight equally between preceding and following positions and normalizes by valid weights at boundaries. Table A.2 specifies the channel counts and radius. In the network descriptions, MLP, FFN, and SiLU denote multilayer perceptron, feed-forward network, and sigmoid linear unit, respectively.

Motion assets. We first pretrain the pose codec, then freeze its parameters and train Root-Weaver. Their objectives are given in Section A.3. Both use Adam and are trained for 500 epochs each, with 120 updates per epoch using sampling with replacement. We multiply each learning rate by 0.1 after epochs 100 and 200. The initial learning rates for the pose codec and Root-Weaver are 3×10^{-5} and 5×10^{-5}, respectively. Subsequent stages use their epoch-500 and epoch-100 checkpoints, respectively. Both use a weight decay of 0, with betas of (0.5,0.999) and (0.9,0.999), respectively.

CWM forward training. With the pose encoder frozen, we jointly train the student music encoder and forward model F for 200 epochs. Each uses a separate AdamW (Loshchilov & Hutter, 2019) optimizer. The student encoder uses betas of (0.9,0.95) and a weight decay of 0.01, with a learning rate that decays from 10^{-4} to 10^{-5} on a cosine schedule. The forward model F uses betas of (0.9,0.999) and a weight decay of 10^{-4}, with a learning rate that decays from 2×10^{-4} to 2×10^{-5} on a cosine schedule. Each optimizer warms up to its specified peak learning rate over the first 5% of updates. We initialize the music teacher from the student and then update it by EMA after every student update. The EMA coefficient increases linearly from 0.99 to 0.9995, and gradients are stopped through the teacher targets.

CWM inverse training. After forward training, we freeze the final music teacher and F, then train H separately for 200 epochs. The inverse model H uses its own AdamW optimizer, with the same optimizer configuration and learning-rate schedule as F. The hidden layers of F and H use SiLU. All three error terms in Eqs. (5a)–(5b) have a weight of 1. We first average MSE over feature dimensions, then over valid adjacent transitions and windows. Both stages exclude bias and LayerNorm parameters from weight decay and clip the gradient norm to 1.

Generator and critic. PCG first undergoes 200 epochs of FM-only pretraining, followed by 300 epochs of L2G-Flow training. This stage uses AdamW with 176 updates per epoch. The learning rate warms up to 2×10^{-5} over the first 880 updates and then decays to 2×10^{-6} on a cosine schedule. We use betas of (0.9,0.999), a weight decay of 0, and gradient-norm clipping at 1. The weights of FM, FK-pos, and ADV are 1, 0.646, and 0.1, respectively. The critic's hinge and R1 coefficients are both 1. In each update, we first detach the generated states from the computation graph and use them to update the critic. We then freeze the critic parameters and update PCG, retaining the gradients of the loss with respect to the generated states. We truncate backpropagation through the sampling rollout every five Euler steps.

Discriminator objective. We use hinge losses (Lim & Ye, 2017) with R_1 regularization (Mescheder et al., 2018). For the generated and matched reference states defined in Section 3.3, the complete discriminator loss is:

$$\mathcal{L}_D = \frac{1}{T}\sum_{k=0}^{T-1}\mathbb{E}\Big[h\big(1 - D_\eta(X_k^{\mathrm{ref}}, \tau_k, c)\big) + h\big(1 + D_\eta(\mathrm{sg}(X_k), \tau_k, c)\big)\Big] + \gamma R_1. \tag{A.6}$$

Here, $h(u) = \max(0, u)$, and sg denotes stop-gradient.

For the Endpoint (final-state ADV) comparison, both the generator's and the discriminator's adversarial losses are averaged over the actual number of supervised states: 1 for endpoint supervision and 20 for per-state supervision. The R1 term retains the same effective regularization strength. Loss weights and gradient truncation follow the configuration above.

Ablation controls. Table 4(a) changes only the temporal mixer. BiMamba shares weights across directions, and Causal Trans. denotes a causal Transformer. Table 4(b) changes only body coordination. All other settings remain fixed in both comparisons. In Table 4(d), only the generative method varies, with DDIM sampling for Diffusion. Only L2G-Flow uses per-state ADV.

Motion conditioning. SAC uses historical pose prompts only, sparse anchors only, both, or no motion cues with probabilities of 45%, 35%, 10%, and 10%, respectively. When anchors are present, we first choose 1, 2, or 3 anchors with probabilities of 70%, 20%, and 10%, respectively, then sample their positions uniformly without replacement from the region to be generated. We drop the music condition with a probability of 10%. The first 25 epochs use clean conditions. Thereafter, historical pose prompts and anchors each receive Gaussian perturbations with a probability of 25%, with standard deviations of 0.02 and 0.05, respectively.

Motion memory. For each dataset, we encode all of its complete training windows and store the 31st token of each window in the memory, retaining its encoding context. Valid anchor positions use zero-based indexing: 1–59 in the first window and 12–59 in subsequent windows. In the default single-anchor configuration, score ties are resolved by selecting the earliest valid position.

Window assembly. A first window contributes L latent tokens. Each subsequent window retains the previous ending's L_p tokens as history and contributes $L - L_p$ new tokens. We concatenate only new regions, decode the assembled poses, and accumulate root displacement along the sequence. With $L = 60$ and $L_p = 12$, subsequent windows contribute 48 new tokens.

B Evaluation Details

B.1 Metrics and Evaluation Clips

We compute both FID and DIV in the kinetic and geometric feature spaces (Li et al., 2021; Siyao et al., 2022; Heusel et al., 2017). Let $q \in \{k, g\}$. The pair (μ_r^q, Σ_r^q) contains the mean and covariance

of the real features, and $\left(\mu_g^q, \Sigma_g^q\right)$ contains those of the generated features. Then:

$$\mathrm{FID}_q = \|\mu_r^q - \mu_g^q\|_2^2 + \mathrm{tr}\Big(\Sigma_r^q + \Sigma_g^q - 2\big((\Sigma_r^q)^{1/2}\Sigma_g^q(\Sigma_r^q)^{1/2}\big)^{1/2}\Big). \tag{B.1}$$

For N generated sequences, let ϕ_q denote the corresponding feature extractor. DIV averages over all pairs of distinct sequences (Li et al., 2021; Siyao et al., 2022):

$$\mathrm{DIV}_q = \frac{2}{N(N-1)} \sum_{1 \le a < b \le N} \|\phi_q(x_a) - \phi_q(x_b)\|_2. \tag{B.2}$$

FSR follows the public implementation of FlowerDance (Yang et al., 2026a) for the foot-physics score (Tseng et al., 2023) and retains the column label used in the main text. For the nth motion sequence with F_n frames, we compute:

$$\mathrm{FSR} = \frac{1}{N}\sum_{n=1}^{N} S_n, \quad S_n = \frac{10^4}{F_n - 2}\sum_{f=1}^{F_n-2} \tilde{a}_{n,f} d_{n,f}^L d_{n,f}^R. \tag{B.3}$$

We compute $\tilde{a}_{n,f}$ in two steps. First, we clamp the vertical component of the second-order differences of root positions to nonnegative values and compute the norm. We then normalize it by the maximum norm within the sequence. The terms $d_{n,f}^L$ and $d_{n,f}^R$ each take the smaller horizontal inter-frame displacement norm of the ankle and toe on the corresponding side.

For each music beat, BAS computes its distance to the nearest motion beat (Li et al., 2021; Siyao et al., 2022; Li et al., 2024):

$$\mathrm{BAS} = \frac{1}{|\mathcal{B}_m|}\sum_{b \in \mathcal{B}_m} \exp\Big(-\frac{(b - b_d)^2}{2\sigma_b^2}\Big). \tag{B.4}$$

Here, $\mathcal{B}_m$ is the set of music beats, and b_d is the motion beat nearest to music beat b. We take $\sigma_b = 3$ frames. Motion beats are identified as local minima of the smoothed mean joint speed. We average BAS across sequences to obtain the set-level score.

B.2 Result Aggregation and Standard Deviations

The main text reports experimental results as means. This appendix provides sample standard deviations (SDs) and aggregation details for ablations (Tables C.8–C.9), user ratings (Table C.6), and long-sequence tests (Figure C.1 and Table C.7).

Ablations and long-duration generation. Unless otherwise specified, we use R=3 independent training runs with different random seeds. Each run undergoes S=5 sampling evaluations over the complete test set. Let $z_{r,s}$ denote the set-level metric from evaluation s of run r. We first average within each run, then compute the mean and sample standard deviation (SD) of these run means:

$$\bar{z}_r = \frac{1}{S}\sum_{s=1}^{S} z_{r,s}, \qquad \bar{z} = \frac{1}{R}\sum_{r=1}^{R} \bar{z}_r, \qquad \mathrm{SD}_{\mathrm{abl}} = \sqrt{\frac{1}{R-1}\sum\nolimits_{r=1}^{R}(\bar{z}_r - \bar{z})^2}. \tag{B.5}$$

For inference-strategy comparisons, variants share one SAC checkpoint within each training run. We pair the initial generation noise by repeat, clip, and window. Random positions, random motions, and evaluator sampling use separate random-number streams. Architecture and motion-representation comparisons use their corresponding matched models and motion assets.

User study. Each participant evaluates all 6 methods on 6 music clips. DQ measures motion naturalness and physical plausibility. DS measures how well the motion coordinates with the music's rhythm and dynamic changes. DC measures motion variation and choreographic expressiveness. Each method receives N=180 raw ratings per dimension. Figure 6 shows their mean as bar height and sample SD as error bars; Table C.6 reports the values. For individual ratings y_n, we compute:

$$\bar{y} = \frac{1}{N}\sum_{n=1}^{N} y_n, \qquad \mathrm{SD}_{\mathrm{user}} = \sqrt{\frac{1}{N-1}\sum\nolimits_{n=1}^{N}(y_n - \bar{y})^2}. \tag{B.6}$$

B.3 Efficiency Measurement

Efficiency evaluations follow FlowerDance (Yang et al., 2026a) in output lengths and hardware settings, as specified in Table 2 of the main text. All ablations use 1,024-frame outputs. We additionally report results for 10,240 frames only in the efficiency comparison between the baselines and full CueDance. In our FPS measurements, the input consists of pre-extracted music features. Timing begins when the conditioning features are ready and ends when the complete motion has been decoded. It includes CWM, retrieval, sampling, decoding, and any necessary waiting, and excludes music-feature extraction from raw audio. For the 1,024-frame benchmark, we run the same set of 84 sequences one at a time.

C Supplementary Analyses

C.1 Supplementary Analysis of 3A

Timing in 3A based on explanatory gain improves both FIDs over fixed and random placement (Table C.1; Main Table 3). The generator, motion-prediction rule, memory, MAI, and single-anchor budget remain fixed; the no-SAC row is a separate training reference. Middle gives the best heuristic FID-k (15.76) and Tail the best heuristic FID-g (17.11). The lower values of 12.39/16.17 with 3A favor timing adapted to the music transition. Although ΔMusic matches GT's FSR, its weaker FIDs and BAS show that this foot-motion statistic alone does not capture the benefit of motion-aware timing.

Explanatory gain measures the contribution of a motion hypothesis to transition prediction. The ΔMusic baseline measures CWM feature change, $\|s_{i+1} - s_i\|_2^2$, while Raw Error measures $d(F_\psi(s_i, 0), s_{i+1})$. The zero-action branch learns to preserve the current state. If $F_\psi(s_i, 0) = s_i$, the scores differ only by the feature-dimension factor and rank positions identically; finite fitting error can change Raw Error's ranking. Our 3A subtracts the action-conditioned next-state residual from Raw Error, measuring the absolute prediction improvement from the motion hypothesis. It therefore prioritizes transitions better explained by that hypothesis over transition change or difficulty alone.

Retrieved examples provide better anchors than direct inverse predictions. Direct Prediction shares full 3A's selection, PCG/CWM, MAI, and paired initial noise, replacing only the anchor value with the inverse prediction at its original standardized scale. Retrieval lowers FID-k/FID-g from 15.40/17.60 to 12.39/16.17. This supports using predicted motion to guide retrieval and retrieved motion as the generation reference.

Table C.1: Additional 3A position-selection comparisons and a no-SAC training reference (mean±SD). The full 3A results appear in Main Table 3.

Configuration	FID-k ↓	FID-g ↓	FSR ↔	DIV-k ↑	DIV-g ↑	BAS ↑
No SAC, no 3A	$19.36^{\pm 0.78}$	$18.11^{\pm 0.63}$	$\underline{0.231}^{\pm 0.007}$	$\mathbf{9.57}^{\pm \mathbf{0.17}}$	$\mathbf{6.94}^{\pm \mathbf{0.19}}$	$\mathbf{0.224}^{\pm \mathbf{0.005}}$
Random Position	$18.86^{\pm 0.84}$	$\underline{17.69}^{\pm 0.75}$	$\mathbf{0.227}^{\pm \mathbf{0.017}}$	$8.52^{\pm 0.25}$	$6.54^{\pm 0.20}$	$\underline{0.222}^{\pm 0.004}$
Middle	$\mathbf{15.76}^{\pm \mathbf{0.67}}$	$17.70^{\pm 0.60}$	$0.193^{\pm 0.012}$	$\underline{8.58}^{\pm 0.18}$	$\underline{6.83}^{\pm 0.16}$	$0.220^{\pm 0.003}$
Tail	$\underline{17.26}^{\pm 0.77}$	$\mathbf{17.11}^{\pm \mathbf{0.74}}$	$0.186^{\pm 0.011}$	$8.22^{\pm 0.22}$	$6.37^{\pm 0.18}$	$0.216^{\pm 0.004}$

Table C.2: FSR for the 3A comparisons in Main Table 3 (mean±SD).

3A setting	FSR ↔	3A setting	FSR ↔
SAC w/o anchor	$0.227^{\pm 0.013}$	Random Action	$0.244^{\pm 0.018}$
Beat/Onset	$0.227^{\pm 0.012}$	NNR	$0.238^{\pm 0.014}$
ΔMusic	$\mathbf{0.216}^{\pm \mathbf{0.016}}$	Direct Prediction	$0.226^{\pm 0.011}$
Raw Error	$0.212^{\pm 0.011}$	Serial 3A	$\underline{0.219}^{\pm 0.009}$

C.2 Number of Anchors

One anchor gives the lowest FIDs and highest diversity among the tested counts. Each anchor is a 254-dimensional token at a four-frame stride: 240 frames become 60 tokens. We rank all valid transitions, independently retrieve one token for each top-4 or top-8 target, and incorporate it through MAI. Compared with top-1's 12.39/16.17, FID-k/FID-g rise to 16.72/18.76 and 17.24/22.31; both diversity scores also fall (Table C.3). These results favor concentrating retrieval on the transition with the highest explanatory gain with the current SAC model, whose anchor-conditioned training uses one to three anchors.

Table C.3: Number of 3A anchors selected by explanatory gain (mean±SD). The single-anchor (top-1) baseline appears in Main Table 3.

Number of Anchors	**FID-k ↓**	**FID-g ↓**	**FSR ↔**	**DIV-k ↑**	**DIV-g ↑**	**BAS ↑**
4(top-4)	$\mathbf{16.72^{\pm 0.83}}$	$\mathbf{18.76^{\pm 0.94}}$	$\mathbf{0.213^{\pm 0.013}}$	$\mathbf{8.21^{\pm 0.21}}$	$\mathbf{5.76^{\pm 0.20}}$	$\underline{0.225}^{\pm 0.004}$
8(top-8)	$\underline{17.24}^{\pm 1.06}$	$\underline{22.31}^{\pm 1.27}$	$\underline{0.248}^{\pm 0.016}$	$\underline{8.17}^{\pm 0.23}$	$\underline{5.65}^{\pm 0.22}$	$\mathbf{0.227^{\pm 0.006}}$

C.3 Pose and Root Displacement

The root-modeling comparison favors recovering displacement from lower-body motion (Table C.4; default in Main Table 4(a)). Joint Root–Lower improves both FIDs over Separate Root, supporting coupled modeling. Root-Weaver's Lower→Root direction further lowers FID-k/FID-g from 17.64/19.26 to 12.39/16.17. Reversing this direction gives the poorest FIDs and an FSR of 0.155, farther from GT's 0.216 than the default's 0.219. Together, these results favor the default direction for motion fidelity and foot-motion statistics.

Table C.4: Root-displacement alternatives to Root-Weaver. FID/BAS are reported as mean±SD, and FSR/DIV as means. The default Lower→Root configuration uses Root-Weaver. Its results appear in the ForeGate row of Main Table 4(a).

Root modeling	**FID-k ↓**	**FID-g ↓**	**FSR ↔**	**DIV-k ↑**	**DIV-g ↑**	**BAS ↑**
Separate Root	$\underline{23.17}^{\pm 0.83}$	$\underline{21.03}^{\pm 0.79}$	$\underline{0.266}$	$\mathbf{9.04}$	$\mathbf{6.82}$	$\mathbf{0.228^{\pm 0.003}}$
Joint Root–Lower	$\mathbf{17.64^{\pm 0.98}}$	$\mathbf{19.26^{\pm 0.84}}$	$\mathbf{0.244}$	$\underline{8.73}$	$\underline{6.67}$	$\underline{0.227}^{\pm 0.003}$
Root→Lower	$27.31^{\pm 1.44}$	$26.17^{\pm 1.14}$	0.155	7.32	5.91	$0.223^{\pm 0.004}$

C.4 Internal Design of ForeGate

ForeGate benefits from separating content aggregation from contextual gating (Table C.5). CC makes both paths causal and raises FID-g from 16.17 to 28.71. MC instead mirrors content and makes gating causal, raising FID-k from 12.39 to 38.88. The default improves both FIDs over either alternative, supporting causal content regulated by context on both sides within the window. Dilation provides a second distinction: increasing a constant rate from 1 to 4 improves FID-k but worsens FID-g. The default schedule, $(1, 2, 4, 1, 2, 4, 1, 2)$, improves both FIDs over every tested constant rate, favoring mixed temporal scales.

Table C.5: ForeGate: content/gate directions and dilation rates (means). The default configuration appears in Main Table 4(a).

ForeGate setting	FID-k ↓	FID-g ↓	FSR ↔	DIV-k ↑	DIV-g ↑	BAS ↑
CC	**14.41**	28.71	0.339	7.38	6.24	<u>0.227</u>
MC	38.88	**19.24**	<u>0.253</u>	<u>10.61</u>	6.13	0.224
Constant d=1	47.38	<u>20.84</u>	**0.207**	**12.39**	6.20	0.226
Constant d=2	21.88	22.24	0.260	8.37	**6.47**	0.225
Constant d=4	<u>16.90</u>	33.27	0.383	7.43	<u>6.32</u>	**0.231**

C.5 Complete Numerical Results

Table C.6: User ratings (mean±SD), corresponding to Figure 6.

Method	DQ ↑	DS ↑	DC ↑
FineNet (2023)	$3.21^{\pm 0.72}$	$3.12^{\pm 0.96}$	$3.25^{\pm 0.92}$
Lodge (2024)	$3.62^{\pm 0.78}$	$3.41^{\pm 0.83}$	$3.72^{\pm 0.81}$
MEGADance (2025b)	$4.02^{\pm 0.83}$	$3.87^{\pm 0.74}$	$3.79^{\pm 0.75}$
FlowerDance (2026a)	$3.73^{\pm 0.74}$	$3.98^{\pm 0.61}$	<u>$4.21^{\pm 0.71}$</u>
MACE-Dance (2026b)	<u>$4.07^{\pm 0.69}$</u>	<u>$4.25^{\pm 0.65}$</u>	$4.07^{\pm 0.69}$
CueDance	$\mathbf{4.50^{\pm 0.65}}$	$\mathbf{4.57^{\pm 0.57}}$	$\mathbf{4.53^{\pm 0.67}}$

CueDance's highest mean quality and synchronization ratings coincide with the lowest raw-rating SDs (0.65/0.57; Table C.6). These judgments are less dispersed on both criteria.

C.6 Long-Sequence Generation

Protocol. Each of the 84 test music clips is looped for ten cycles, with each cycle aligned to 1,024 motion frames at 30 FPS (approximately 341 s in total). Motion is generated continuously from the repeated music. The Pose Prompt carries over across generation windows and music-loop boundaries, preserving history and allowing preceding errors to propagate. Within each training run, both configurations share a SAC checkpoint; retrieval anchors remain enabled in every window or disabled throughout. We evaluate the final 1,024 generated frames after 1, 2, 4, 6, 8, and 10 cycles, using each configuration's first segment as its retention reference. Reported approximate durations use 34.13 s per cycle.

Our 3A improves relative quality retention at every measured subsequent time point (Figure C.1). The two configurations share a SAC checkpoint and differ in whether retrieval anchors are enabled. Each index normalizes a configuration to its own first segment. At 341.30 s, anchors raise geometric/kinetic retention from 0.660/0.422 to 0.823/0.785. The kinetic index exceeds 1 at 68.26 s because that segment improves on its own initial segment. This sustained relative advantage supports repeated reference access during long-sequence generation.

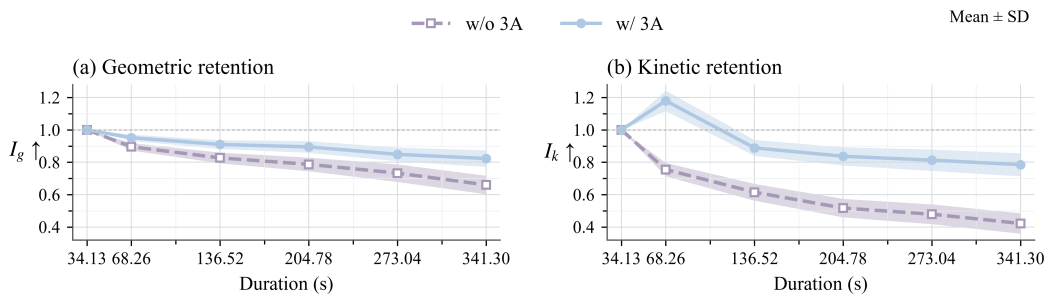


Figure C.1: Long-sequence relative quality retention. Both groups share the same SAC checkpoint. w/o and w/ 3A disable and enable retrieval anchors, respectively. Curves and shading show the mean and ±1 SD across three training-run means. We compute FID over the final 1,024 frames at each time point and normalize it as each configuration's first-segment FID/current-segment FID. Table C.7 gives the complete values.

Table C.7: Long-sequence relative quality retention with and without 3A (mean±SD), corresponding to Figure C.1.

Approx. Duration (s)	I_g **w/o 3A**	I_g **w/ 3A**	I_k **w/o 3A**	I_k **w/ 3A**
34.13	$1.000^{\pm 0.000}$	$1.000^{\pm 0.000}$	$1.000^{\pm 0.000}$	$1.000^{\pm 0.000}$
68.26	$0.896^{\pm 0.023}$	$0.952^{\pm 0.019}$	$0.755^{\pm 0.041}$	$1.180^{\pm 0.063}$
136.52	$0.827^{\pm 0.033}$	$0.910^{\pm 0.027}$	$0.615^{\pm 0.052}$	$0.889^{\pm 0.049}$
204.78	$0.787^{\pm 0.042}$	$0.895^{\pm 0.036}$	$0.517^{\pm 0.057}$	$0.837^{\pm 0.056}$
273.04	$0.733^{\pm 0.054}$	$0.849^{\pm 0.042}$	$0.479^{\pm 0.061}$	$0.813^{\pm 0.066}$
341.30	$0.660^{\pm 0.057}$	$0.823^{\pm 0.051}$	$0.422^{\pm 0.063}$	$0.785^{\pm 0.071}$

C.7 Variability of Ablation Results

Tables C.8 and C.9 provide sample standard deviations for the reported metrics in Main Tables 3 and 4, in the same order. Section B.2 defines aggregation for these ablations and the long-sequence results.

Table C.8: 3A: SDs for Table 3.

3A setting	**FID-k**	**FID-g**	**DIV-k**	**DIV-g**	**BAS**
When: Reference timing					
Beat/Onset	0.61	0.61	0.21	0.13	0.002
ΔMusic	0.78	0.72	0.19	0.19	0.006
Raw Error	0.71	0.63	0.23	0.12	0.003
What: Reference content					
SAC w/o anchor	0.83	0.72	0.27	0.12	0.006
Random Action	1.17	1.03	0.28	0.22	0.004
Direct Prediction	0.90	0.60	0.22	0.19	0.003
Retrieval and execution					
NNR	0.87	0.78	0.26	0.17	0.006
Serial 3A	0.54	0.59	0.16	0.22	0.002
3A	0.54	0.57	0.16	0.15	0.002

Table C.9(a): ForeGate: SDs for Table 4(a).

Temporal mixer	FID-k	FID-g	FSR	DIV-k	DIV-g	BAS
Causal Trans.	0.83	0.85	0.012	0.21	0.16	0.003
BiMamba	0.97	1.17	0.014	0.22	0.18	0.003
Mamba	0.79	1.25	0.018	0.24	0.19	0.004
ForeGate	0.54	0.57	0.007	0.16	0.15	0.002

Table C.9(b): Unison Hint: SDs for Table 4(b).

Body coord.	FID-k	FID-g	FSR	DIV-k	DIV-g	BAS
Joint attention	1.17	1.07	0.012	0.27	0.18	0.004
Cross attention	1.13	1.02	0.010	0.23	0.18	0.005
Unison Hint	0.54	0.57	0.007	0.16	0.15	0.002

Table C.9(c): L2G-Flow supervision: SDs for Table 4(c).

Objective	FID-k	FID-g	FSR	DIV-k	DIV-g	BAS
FM	1.54	1.83	0.016	0.23	0.32	0.005
FM+ADV	0.83	0.92	0.009	0.22	0.18	0.003
FM+FK-pos	1.24	1.33	0.013	0.27	0.22	0.004
FM+FK-pos+Endpoint	1.10	0.90	0.011	0.23	0.22	0.005
L2G-Flow	0.54	0.57	0.007	0.16	0.15	0.002

Table C.9(d): L2G-Flow comparison: SDs for Table 4(d).

Method	FID-k	FID-g	FSR	DIV-k	DIV-g	BAS
Rectified Flow (3× reflow)	0.80	0.83	0.012	0.22	0.16	0.003
Diffusion	0.93	0.90	0.016	0.26	0.17	0.004
MeanFlow	0.96	1.01	0.011	0.19	0.18	0.003
L2G-Flow	0.54	0.57	0.007	0.16	0.15	0.002

Variation across training runs. With FM and FK-pos fixed, per-state ADV improves mean FIDs over Endpoint while lowering their training-run SDs from 1.10/0.90 to 0.54/0.57 (Table C.9(c)). Thus, the quality gain accompanies lower observed variability. Serial and parallel 3A have similar SDs (0.54/0.59 versus 0.54/0.57), and their mean FIDs differ by only 0.01/0.01. These descriptive results support similar observed quality under the scheduling change; the measured throughput gain appears in Main Table 3.

C.8 Local Correction Versus Propagation

Table C.10: Local correction versus propagation on FineDance. In each single-window control, 3A anchors a single window while all other windows remain anchor-free. Window 5 is the last complete window within 1,024 frames, with only 16 evaluated frames following that window. All rows come from a single training run with the same SAC checkpoint and paired initial noise.

Anchor setting	FID-k↓	FID-g↓	DIV-k↑	DIV-g↑	BAS↑
No anchors	20.78	18.29	9.42	6.84	0.221
Window 5 only	19.61	18.23	9.33	6.81	0.224
Window 1 only	17.73	17.62	9.17	6.72	0.226
Every window (3A)	12.39	16.17	8.94	6.71	0.235