WHERE THE BODY KEEPS THE BEAT: STRUCTURED MOTION CONDITIONING AND MUSIC DYNAMICS SUPERVISION FOR DANCE-TO-MUSIC GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dance-to-music (D2M) generation aims to synthesize musically plausible soundtracks whose temporal structure aligns with a given dance performance. Representative D2M methods do not explicitly distinguish motion cues across body parts and frequency bands, while standard flow matching lacks a dedicated objective for supervising local music-latent dynamics. To address these issues, we propose **Dyna2Music**, a latent flow-matching framework that combines structured motion conditioning with explicit supervision of music-latent dynamics. An empirical analysis on AIST++ quantifies how spatial partitioning and frequency separation affect raw music-to-kinematic beat alignment and motion-reference density, informing the conditioning design. Accordingly, Dyna2Music decomposes joint velocities into slow and fast components and hierarchically fuses the resulting part-wise motion energy with pretrained joint features to condition music generation. To complement this representation, we introduce latent dynamics consistency (LDC), an auxiliary objective that matches adjacent-frame change magnitudes between a single-step clean-latent estimate and the paired reference. LDC makes local music-latent variation an explicit training target without adding trainable parameters or inference computation. Dyna2Music supports variable-length music generation, and experiments on AIST++ and TikTok demonstrate improved rhythmic alignment and audio quality over representative D2M baselines.

1 INTRODUCTION

For dance videos shared on platforms such as TikTok and YouTube, the choice of background music shapes both the expressive quality of the performance and the viewing experience. This application motivates research into dance-to-music (D2M) generation (Gan et al., 2020; Di et al., 2021; Zhu et al., 2022a;b; Yu et al., 2023; Han et al., 2024; Li et al., 2024; Zhang & Hua, 2024; Liang et al., 2024; Sun et al., 2025; Ji et al., 2025). Unlike music generation for general video content, which addresses audiovisual compatibility through visual-semantic conditioning (Tian et al., 2025) and rhythmic controls derived from visual motion speed and saliency (Di et al., 2021), D2M places particular emphasis on synthesizing musically plausible soundtracks whose temporal structure aligns with a given dance performance. Beyond matching the overall style or tempo, the soundtrack should respond to changes in movement, including sustained body motion, brief limb articulations, and rhythmic accents. Addressing these requirements involves both representing dance dynamics and learning the temporal structure of the accompanying music.

Existing diffusion-based D2M approaches typically condition music generation on visual features and pose-derived motion cues extracted from dance videos. Within this paradigm, LORIS (Yu et al., 2023) derives a single visual rhythm sequence from aggregated joint motion, while PN-Diffusion (Sun et al., 2025) uses a spatiotemporal graph convolutional network to encode skeletal sequences into a motion embedding. However, the resulting conditioning representations do not explicitly separate cues by body part or temporal frequency. *Spatially*, different body parts may provide complementary cues about musical beat timing. Aggregating motion across parts can obscure localized events, such as a brief wrist accent occurring alongside a larger torso movement, making these cues less distinguishable in the conditioning representation. To examine this issue, we analyze $8{,}819$ paired

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.

training clips from AIST++ (Li et al., 2021), a dataset of dance videos paired with music that is widely used as a benchmark for D2M generation (Zhu et al., 2022b; Yu et al., 2023; Sun et al., 2025). Figure 1 compares whole-body, two-part, and five-part motion references, each formed by taking the union of motion-derived beat candidates (kinematic beats) from its constituent signals. We use the music-to-kinematic beat alignment score (BAS) (Siyao et al., 2022) to measure each music beat's temporal proximity to its nearest kinematic beat. Without frequency separation, raw BAS increases from 0.307 to 0.418 and 0.569 as the spatial partition becomes finer. *Temporally*, slowly varying dynamics and rapid local fluctuations may convey complementary information about musical tempo, beat phase, and metrical subdivisions. Although existing methods capture temporal dependencies, they do not explicitly represent low-frequency (slow) and high-frequency (fast) motion components as distinct conditioning signals. Separating slow and fast components further increases raw BAS to 0.472 and 0.661 for the two- and five-part references, respectively; the five-part reference thus scores higher in both settings. These raw scores also depend on motion-reference density, as illustrated in Figure 1. We therefore investigate how part-specific slow and fast motion signals relate to musical rhythm, using the findings to inform the conditioning design.

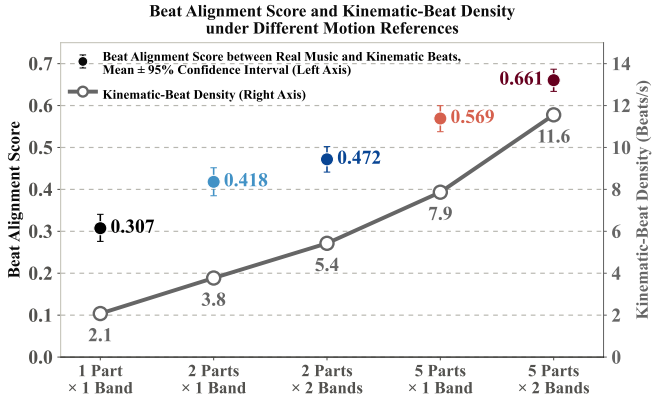


Figure 1: Beat alignment score (BAS) and kinematic-beat density for different motion references on paired dance-music clips from the AIST++ training set. Filled markers and error bars (left axis) indicate mean music-to-kinematic BAS and 95% confidence intervals, respectively; the grey line (right axis) shows kinematic-beat density (beats/s). The spatial partitions comprise the whole body (one part), upper and lower body (two parts), or the torso, left and right arms, and left and right legs (five parts). One band denotes motion without frequency separation; two bands denote separate slow and fast components. Finer spatial partitions and slow-fast separation yield denser motion references, helping to explain the increase in raw BAS.

Beyond representing rhythmic cues in dance, D2M generation also depends on how temporal variation in the accompanying music is learned. For generators operating in audio latent space, the magnitudes of changes between adjacent latent frames characterize local temporal variation in the audio representation. The standard flow-matching objective (Lipman et al., 2022) regresses the generative velocity field without a dedicated term for comparing these dynamics between the predicted clean latent and the paired reference. We therefore augment velocity regression with an auxiliary objective that matches these change magnitudes at corresponding music-latent frames. This provides explicit supervision of local audio-latent dynamics, complementing part-specific slow and fast motion conditioning.

To this end, we introduce **Dyna2Music**, a latent flow-matching framework for D2M generation that combines structured motion conditioning with explicit supervision of music-latent dynamics (Figure 2). To preserve motion cues across body parts and temporal scales, the **Spatio-Spectral Motion Decomposition (SSMD)** module separates joint velocities into low- and high-frequency components and computes their energy for each of five body parts. These energy signals are complemented by pretrained MotionBERT features (Zhu et al., 2023), pooled within each part to provide information about joint configurations and coordination that energy alone does not capture. The **Hierarchical Slow-Fast Conditioning Encoder (HSF-Encoder)** then integrates the energy signals and motion features through attention across body parts and adaptive gating, producing conditioning tokens that combine part-specific context with slow and fast motion dynamics. The resulting conditioning tokens guide a transformer adapted from DiffRhythm (Ning et al., 2025) through cross-attention to generate continuous music latents, which are decoded into music by a frozen waveform VAE. To complement structured motion conditioning, we introduce **latent dynamics consistency (LDC)** as an auxiliary objective for supervising music-latent dynamics. During training, we derive a single-step estimate of the clean music latent from the predicted velocity field. LDC matches the magnitudes of adjacent-frame changes in this estimate to those in the paired reference

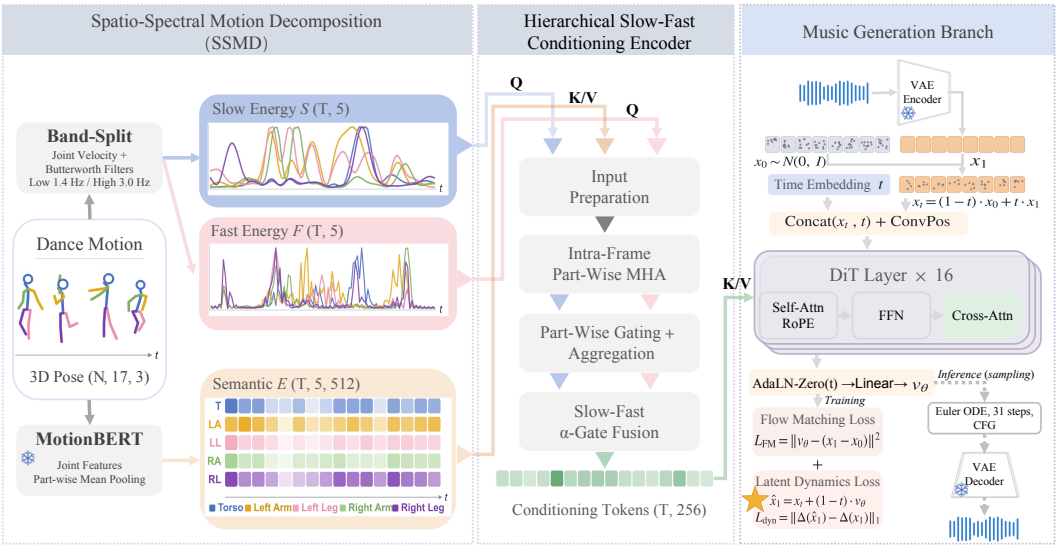


Figure 2: **Overview of Dyna2Music. Left:** SSMD extracts part-wise slow/fast motion energy and complementary features from frozen MotionBERT. **Middle:** The HSF-Encoder fuses these signals into conditioning tokens through attention and adaptive gating. **Right:** The tokens guide a latent flow-matching transformer trained with velocity regression and latent dynamics consistency (LDC), which supervises adjacent-frame change magnitudes. Generated latents are decoded into music by a frozen waveform VAE.

latent at corresponding music time steps. This explicitly supervises the temporal profile of local latent variation, complementing the standard flow-matching objective. The loss is computed entirely in latent space, requires no waveform decoding during training, and introduces neither additional trainable parameters nor additional computation at inference. We summarize our contributions as follows.

- **An empirical study of motion references.** We quantify how body-part partitioning and slow-fast separation affect raw music-to-kinematic beat alignment and kinematic-beat density on AIST++, informing the design and evaluation of motion conditioning.
- **Structured motion conditioning for D2M generation.** We propose Dyna2Music, a latent flow-matching framework that combines explicit motion decomposition across body parts and frequency bands with hierarchical fusion of motion energy and pretrained joint features.
- **Explicit supervision of music-latent dynamics.** We introduce LDC, an auxiliary objective that matches adjacent-frame change magnitudes between a single-step clean-latent estimate and the paired reference. This complements velocity regression with explicit supervision of local temporal variation, without additional trainable parameters or inference computation.
- **Variable-length generation and empirical validation.** Dyna2Music supports music generation for dance sequences of varying durations. Experiments on AIST++ and TikTok demonstrate improved rhythmic alignment and audio quality over representative D2M baselines.

2 Related Work

General Video-to-Music Generation. It targets background music generation for diverse content, including films, vlogs, and documentaries, emphasizing semantic compatibility, temporal alignment, and controllability. An early approach, CMT (Di et al., 2021), controls symbolic music generation by mapping visual timing, motion speed, and motion saliency to beats, note density, and note strength. For richer visual context, VidMuse (Tian et al., 2025) integrates local clip information and global video context through long-short-term modeling, while GVMGen (Zuo et al., 2025) selects visual features with learnable music queries and conditions generation through temporal cross-attention. For explicit control, MTCV2M (Wu et al., 2025) combines fine-grained feature selection, progressive temporal alignment, and dynamic fusion of time-varying beat, melody, intensity, and emotion conditions. Complementary approaches use pretrained multimodal and language models to link video understanding with music generation. M^{2}UGen (Liu et al., 2023) integrates modality-

specific encoders with LLaMA 2 to condition MusicGen or AudioLDM 2 on inputs including videos. Similarly, SONIQUE (Zhang & Fuentes, 2025) converts video descriptions into musical tags for conditional diffusion, enabling soundtracking with unpaired video and music data. More broadly, AudioX (Tian et al., 2026) conditions a diffusion transformer on fused text, video, and audio representations, supporting video-to-music generation in a unified multimodal framework. These studies explore learned visual representations, explicit musical controls, and language interfaces for general video soundtracking.

Autoregressive Dance-to-Music Generation. Autoregressive D2M methods sequentially predict symbolic events or discrete audio tokens conditioned on motion and preceding outputs. Early approaches include Dance2Music (Aggarwal & Parikh, 2021), which generates notes online from pose and note histories, and RhythmicNet (Su et al., 2021), which converts motion-derived rhythm into percussion and melodic accompaniment. For multi-instrument generation, Dance2MIDI (Han et al., 2024) combines autoregressive drum generation with BERT-based track completion, while DanceComposer (Liang et al., 2024) progressively introduces rhythm and style constraints. Uni-MuMo (Yang et al., 2025) extends this paradigm to unified text, music, and motion generation through a shared music-motion codebook and mutually conditioned autoregressive streams, supporting motion-to-music generation. Across representations, decoding retains sequential dependencies between output time steps.

Non-autoregressive Dance-to-Music Generation. Non-autoregressive D2M methods use adversarial synthesis or iterative refinement without left-to-right token decoding. D2M-GAN (Zhu et al., 2022a) conditions vector-quantized adversarial generation on video and motion, whereas CDCD (Zhu et al., 2022b) introduces contrastive learning into discrete diffusion. In continuous latent space, Tan et al. (2023) incorporate motion and genre conditions, while Dance2Music-Diffusion (Zhang & Hua, 2024) uses a dedicated motion encoder. For richer conditioning, LORIS (Yu et al., 2023) integrates visual, rhythmic, and genre context for long-term synthesis, while PN-Diffusion (Sun et al., 2025) uses positive and negative rhythmic signals from forward and reversed dance sequences. To adapt pretrained diffusion models, encoder-based textual inversion (Li et al., 2024) maps dance rhythm and genre to text embeddings, whereas Kimura et al. (2026) combine beat-guided contrastive pretraining with a ControlNet adapter for a frozen AudioLDM backbone. Within this family, Dyna2Music combines part-specific slow-fast motion conditioning with explicit LDC supervision of local music-latent dynamics in a latent flow-matching framework.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

Dyna2Music conditions music generation on the dancer's motion, represented as a sequence of estimated 3D joint positions $\mathbf{J}$ spanning N video frames. The objective is to synthesize a musically plausible accompaniment whose temporal structure aligns with the observed dance dynamics. We formulate this task as conditional generation in music latent space. For each dance-music pair, a frozen pretrained waveform VAE (Ning et al., 2025) encodes the music waveform $\mathbf{y}$ as

$$\mathbf{x}_1 = \mathcal{E}(\mathbf{y}) \in \mathbb{R}^{T \times d_a}, \tag{1}$$

where T denotes the number of music-latent frames and d_a is the latent dimensionality. We seek to learn the conditional distribution $p_\theta(\mathbf{x}_1 \mid \mathbf{J})$. At inference, generated music latents are converted into waveforms by the frozen VAE decoder $\mathcal{D}$. Throughout this section, i indexes video frames, n indexes music-latent frames, and $t \in [0, 1]$ denotes flow time.

3.2 SPATIO-SPECTRAL MOTION DECOMPOSITION

SSMD constructs motion-energy signals and joint features for the five-part partition examined in Figure 1: the torso, left and right arms, and left and right legs. Let $\mathcal{P}$ denote these parts and $\mathcal{J}_p$ the joint index set of part p within the 17-joint skeleton.

Part-specific motion energy. We compute joint velocities from consecutive poses and apply temporal low-pass and high-pass filters with cutoff frequencies of $1.4\,\text{Hz}$ and $3.0\,\text{Hz}$ to obtain slow and fast velocity sequences $\mathbf{v}^s$ and $\mathbf{v}^f$, respectively. For band $b \in \{s, f\}$, we average the squared speed of

the joints within each part:

$$K_{i,p}^{b} = \frac{1}{|\mathcal{J}_p|} \sum_{j \in \mathcal{J}_p} \|\mathbf{v}_{i,j}^{b}\|_2^2. \tag{2}$$

Frequency separation is applied to joint velocities before energy computation and aggregation within each body part. Here, slow and fast refer to the temporal frequency of velocity variations, rather than the magnitude of joint velocity.

Joint features and temporal alignment. A frozen MotionBERT backbone (Zhu et al., 2023) processes the full-body pose sequence to extract 512-dimensional joint features, which we average within the same parts to provide complementary motion context. We resample the energy signals and pooled features to the T music-latent frames, obtaining $\mathbf{S}, \mathbf{F} \in \mathbb{R}^{T\times 5}$ and $\mathbf{E} \in \mathbb{R}^{T\times 5\times 512}$ as inputs to the HSF-Encoder. The joint partition, filtering, and preprocessing details are specified in Appendix A.1.

3.3 Hierarchical Slow-Fast Conditioning Encoder

As shown in Figure 2, the HSF-Encoder maps $\mathbf{S}$, $\mathbf{F}$, and $\mathbf{E}$ to a conditioning sequence $\mathbf{C} \in \mathbb{R}^{T\times d}$, where $d = 256$. At each music-latent time step, the encoder retains distinct tokens for each part and band during context integration, uses energy-dependent gates to aggregate parts within each band, and then fuses the slow and fast representations.

Input Preparation. Energy sequences are standardized over valid time steps independently for each part and band, while joint features undergo layer normalization. Both are then projected to d dimensions and augmented with temporal and part embeddings; additional band embeddings distinguish the slow and fast energy streams. The resulting representations are denoted by $\widetilde{\mathbf{B}}^{s}$, $\widetilde{\mathbf{B}}^{f}$, and $\widetilde{\mathbf{E}}$. The embedding definitions and normalization details are provided in Appendix A.2; affine biases are omitted from the notation for brevity.

Intra-Frame Part-Wise MHA. At each music-latent frame n, multi-head attention (MHA) integrates joint-feature context across the five body parts using band-specific energy embeddings as queries:

$$\mathbf{H}_{n}^{b} = \mathrm{MHA}_{b}\Big(\widetilde{\mathbf{B}}_{n}^{b}, \widetilde{\mathbf{E}}_{n}, \widetilde{\mathbf{E}}_{n}\Big) \in \mathbb{R}^{5\times d}. \tag{3}$$

The three arguments denote queries, keys, and values, respectively. Query projections are band-specific, whereas key, value, and output projections are shared across bands. This design allows each part's band-specific energy representation to retrieve motion context from all body parts before aggregation.

Part-Wise Gating + Aggregation. For each band, a gating network maps the energy embeddings to scores that are normalized across parts. The resulting weights aggregate the attention outputs into a frame-level representation:

$$g_{n,p}^{b} = \mathrm{softmax}_{p}\Big(\mathrm{MLP}_{b}^{\mathrm{gate}}(\widetilde{\mathbf{B}}_{n,p}^{b})\Big), \qquad \mathbf{h}_{n}^{b} = \sum_{p\in\mathcal{P}} g_{n,p}^{b}\mathbf{H}_{n,p}^{b}. \tag{4}$$

Slow-Fast α-Gate Fusion. A learned scalar gate α_n controls the relative contributions of the slow and fast representations to the mixture $\mathbf{m}_n$. The final projection combines this mixture with their normalized concatenation $\mathbf{u}_n$:

$$\begin{aligned} \mathbf{u}_n &= \mathrm{LN}([\mathbf{h}_n^{s}; \mathbf{h}_n^{f}]), & \alpha_n &= \sigma(\mathbf{w}_{\alpha}^{\mathsf{T}}\mathbf{u}_n), \\ \mathbf{m}_n &= \alpha_n\mathbf{h}_n^{s} + (1-\alpha_n)\mathbf{h}_n^{f}, & \mathbf{C}_n &= \mathrm{LN}\big(\mathbf{W}_F[\mathbf{m}_n; \mathbf{u}_n] + \mathbf{e}_n^{\mathrm{time}}\big). \end{aligned} \tag{5}$$

Retaining the concatenated branch representations alongside their gated mixture gives the final projection access to both branches separately. Here, LN denotes layer normalization, σ is the sigmoid function, $[\cdot;\cdot]$ denotes concatenation, and $\mathbf{e}_n^{\mathrm{time}}$ is the temporal positional embedding. The resulting sequence $\mathbf{C}$ provides the motion conditions for the music generator.

3.4 Latent Flow Matching with Dynamics Supervision

Conditioned on the HSF sequence $\mathbf{C}$, a transformer adapted from DiffRhythm (Ning et al., 2025) predicts the music-latent velocity $\mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{C})$ (Figure 2). In each block, music representations query $\mathbf{C}$ through cross-attention; architecture details are provided in Appendix A.3.

Conditional flow matching. Following conditional flow matching (Lipman et al., 2022), we interpolate Gaussian noise $\mathbf{x}_0$ with the paired music latent $\mathbf{x}_1$ to define the path and target velocity:

$$\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1, \qquad \mathbf{v}^\star = \mathbf{x}_1 - \mathbf{x}_0. \tag{6}$$

The flow-matching loss $\mathcal{L}_{\mathrm{FM}}$ averages squared errors between predicted and target velocities over valid frames and channels. Flow-time sampling and the masked loss reduction are specified in Appendix A.4.

Latent dynamics consistency. LDC complements velocity regression by explicitly supervising local temporal variation in music latents. It uses a single-step estimate of the clean music latent:

$$\widehat{\mathbf{x}}_1 = \mathbf{x}_t + (1-t)\mathbf{v}_\theta(\mathbf{x}_t, t, \mathbf{C}). \tag{7}$$

For a latent sequence $\mathbf{z}$, the adjacent-frame change magnitude is the root mean square across channels:

$$\Delta_n(\mathbf{z}) = \sqrt{\frac{1}{d_a}\sum_{k=1}^{d_a}(z_{n,k} - z_{n-1,k})^2 + \varepsilon}, \qquad n \geq 2, \quad \varepsilon = 10^{-6}. \tag{8}$$

Let q index minibatch samples and $\mathcal{I}_\Delta$ contain pairs (q, n) for which both $n-1$ and n are valid. The auxiliary loss and full objective are

$$\mathcal{L}_{\mathrm{LDC}} = \frac{\sum_{(q,n)\in\mathcal{I}_\Delta} |\Delta_n(\widehat{\mathbf{x}}_{1,q}) - \Delta_n(\mathbf{x}_{1,q})|}{\max(1, |\mathcal{I}_\Delta|)}, \qquad \mathcal{L} = \mathcal{L}_{\mathrm{FM}} + \lambda_{\mathrm{LDC}}\mathcal{L}_{\mathrm{LDC}}. \tag{9}$$

Here, $\lambda_{\mathrm{LDC}} \geq 0$ controls the auxiliary supervision, with zero recovering flow matching alone. LDC matches local change magnitudes at corresponding music frames; it does not directly constrain change directions or beat locations. It uses the same velocity prediction and adds neither trainable parameters nor inference computation. The HSF-Encoder and music transformer are trained jointly, while MotionBERT and the waveform VAE remain frozen.

Inference and variable-length generation. The dance duration and VAE temporal resolution determine the latent length T within the configured sequence-length limit. We integrate the conditional velocity field from Gaussian noise using Euler updates and decode the resulting latents with the frozen VAE. Sampling details are given in Appendix A.4.

4 Experiments

4.1 Datasets

We evaluate Dyna2Music on two complementary D2M benchmarks, AIST++ (Li et al., 2021; Tsuchida et al., 2019) and TikTok (Zhu et al., 2022a), covering dance performances captured in controlled studio environments and in-the-wild social-media videos, respectively. Together, they allow us to assess rhythmic alignment and audio quality across different recording conditions. (i) **AIST++**. This dataset contains dance performances recorded in a controlled studio environment. Our split comprises $8{,}819$ training and 180 test dance video-music pairs. (ii) **TikTok**. This dataset contains in-the-wild dance videos collected from social media, with diverse backgrounds and recording conditions. Our split comprises 391 training and 53 test dance video-music pairs.

Training-length eligibility and sequence caps are detailed in Appendix A.3.

4.2 Baselines and Evaluation Protocols

Baselines. We organize the comparison methods into two groups. *(i) General video-to-music generation:* CMT (Di et al., 2021), M^{2}UGen (Liu et al., 2023), VidMuse (Tian et al., 2025), GVMGen (Zuo et al., 2025), MTCV2M (Wu et al., 2025), SONIQUE (Zhang & Fuentes, 2025), and AudioX (Tian et al., 2026) represent general video soundtracking approaches. The objective comparisons additionally include Diff-Foley (Luo et al., 2023) as a video-to-audio reference. *(ii) Dance-to-music generation:* D2M-GAN (Zhu et al., 2022a), CDCD (Zhu et al., 2022b), LORIS (Yu et al., 2023), Dance2MusicTI (Li et al., 2024), PN-Diffusion (Sun et al., 2025), UniMuMo (Yang et al., 2025), and AudioLDM-ControlNet (Kimura et al., 2026) provide comparisons with methods that condition on human motion, where UniMuMo is evaluated through its motion-to-music interface.

Table 1: Onset-level synchrony (%) on AIST++ and TikTok. BCS/BHS denote precision/recall; CSD/HSD are their across-clip standard deviations. Best values are bold.

Category	Method	AIST++					TikTok				
		BCS↑	CSD↓	BHS↑	HSD↓	F1↑	BCS↑	CSD↓	BHS↑	HSD↓	F1↑
General video-to-music generation	CMT	32.77	23.43	10.51	8.74	14.80	20.20	18.43	12.01	11.54	13.64
	Diff-Foley	31.02	23.10	12.85	9.51	16.80	23.21	20.73	7.86	**9.22**	9.72
	M^{2}UGen	32.10	20.81	43.19	25.70	33.24	19.38	11.91	40.64	25.83	24.71
	VidMuse	33.41	18.41	49.72	17.41	38.06	19.33	**10.12**	42.64	15.79	25.37
	GVMGen	28.18	27.77	12.10	16.35	12.49	18.54	21.56	10.60	11.78	10.82
	MTCV2M	19.66	30.66	4.11	8.32	5.57	17.88	20.29	16.48	14.49	13.73
	SONIQUE	37.36	34.44	8.55	12.60	11.16	22.69	25.51	19.40	17.95	16.16
	AudioX	30.82	25.58	6.93	**6.05**	10.63	18.94	11.47	26.81	16.35	20.12
Dance-to-music generation	D2M-GAN	34.75	**17.50**	31.07	10.68	30.30	18.66	11.43	17.99	9.66	17.28
	CDCD	37.27	24.84	36.88	22.09	30.79	22.01	14.82	25.42	12.37	18.06
	LORIS	33.57	20.38	26.56	14.20	27.36	20.36	14.73	26.51	15.45	20.97
	Dance2MusicTI	32.99	18.47	39.32	20.45	33.79	20.98	11.63	39.93	18.10	25.63
	PN-Diffusion	34.01	18.75	32.44	22.01	29.46	14.42	25.55	5.82	10.08	6.99
	UniMuMo	30.69	26.91	11.16	11.88	13.93	17.56	17.79	11.96	13.06	11.39
	AudioLDM-ControlNet	36.39	24.74	15.67	11.81	20.11	20.30	12.84	19.05	10.67	18.23
	Dyna2Music (Ours)	**60.44**	26.57	**63.54**	25.70	**55.76**	**24.06**	11.44	**44.19**	15.08	**29.99**

Evaluation Protocols. *(a) Objective Evaluation.* We assess onset-level synchrony and audio quality on both datasets. Because LORIS's one-second onset bins obscure fine timing differences, we match onset timestamps one-to-one within ± 50 ms. We report Beats Coverage Score (BCS), Beats Hit Score (BHS), and F1; definitions and across-clip variability statistics (CSD/HSD) are detailed in Appendix B.1. Audio quality is measured by Fréchet Audio Distance (FAD) (Kilgour et al., 2018) using VGGish (Hershey et al., 2017) (FAD$_v$), PANNs (Kong et al., 2020) (FAD$_p$), and CLAP (Elizalde et al., 2023) (FAD$_c$) features, with lower values indicating closer agreement with reference music distributions.

Table 2: FAD (lower is better) on AIST++ and TikTok using VGGish (v), PANNs (p), and CLAP (c) features. Best and second-best values are bold and underlined, respectively.

Category	Method	AIST++			TikTok		
		FAD$_v$↓	FAD$_p$↓	FAD$_c$↓	FAD$_v$↓	FAD$_p$↓	FAD$_c$↓
General video-to-music generation	CMT	10.672	68.863	0.958	18.556	80.538	1.368
	Diff-Foley	31.843	184.737	1.606	15.147	96.495	1.318
	M^{2}UGen	<u>4.081</u>	41.447	0.808	11.493	60.614	1.051
	VidMuse	4.778	<u>28.645</u>	0.975	9.930	52.895	1.008
	GVMGen	12.694	56.269	1.132	10.474	54.344	1.147
	MTCV2M	15.695	62.076	1.130	9.230	61.119	0.952
	SONIQUE	8.203	68.007	1.078	11.048	70.952	1.116
	AudioX	23.151	96.082	1.748	8.539	52.205	0.919
Dance-to-music generation	D2M-GAN	11.719	71.355	1.673	12.885	99.451	1.863
	CDCD	7.478	51.817	1.258	6.532	<u>33.629</u>	1.116
	LORIS	10.733	30.998	1.047	15.872	59.243	1.214
	Dance2MusicTI	8.968	49.539	1.171	10.146	66.981	1.346
	PN-Diffusion	4.832	38.041	<u>0.600</u>	16.729	71.432	1.291
	UniMuMo	10.307	56.046	1.157	11.505	66.003	1.233
	AudioLDM-ControlNet	5.632	29.587	0.677	<u>6.443</u>	41.826	<u>0.771</u>
	Dyna2Music (Ours)	**2.925**	**21.259**	**0.562**	**5.342**	**32.204**	**0.493**

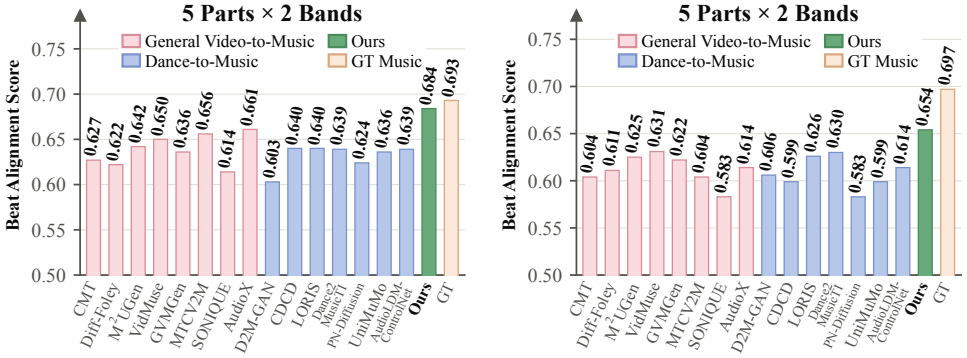
(b) Subjective Evaluation. An AIST++ user study assesses overall musical quality (OVL) and dance-music relevance (REL). The study protocol and results are provided in Appendix E. The supplementary material includes selected generation results from the AIST++ and TikTok test sets.

Implementation Details. Dyna2Music operates on 64-channel music latents from a frozen waveform VAE with a temporal downsampling factor of 2048 and stereo output at 44.1 kHz. The HSF-Encoder and flow-matching transformer are optimized jointly, while MotionBERT and the waveform VAE remain frozen. We use AdamW with a learning rate of 7.5×10^{-5}, linear warm-up followed by linear decay, FP16 mixed precision, and gradient clipping at a norm of 1.0. Training is configured for up to 100 epochs with random seed 42, and conditioning dropout is set to 0.2. We set $\lambda_{\mathrm{LDC}} = 1$ on both datasets. At inference, we use exponential-moving-average weights and Euler integration over 32 uniformly spaced flow-time points. Architecture, sequence-length, and sampling details are provided in Appendix A. The influence of the LDC weight on onset-level synchrony and audio quality is examined in Appendix D.

4.3 Model Comparison

Tables 1 and 2 compare onset-level synchrony and audio quality with fifteen baselines on AIST++ and TikTok. **(i) Onset synchrony.** Dyna2Music achieves the highest BCS, BHS, and F1 on both datasets. Its F1 scores of 55.76 on AIST++ and 29.99 on TikTok exceed the strongest baselines, VidMuse and Dance2MusicTI, by 17.70 and 4.36 percentage points, respectively. The gains in both precision and recall indicate improved agreement with paired onsets, with a larger advantage on AIST++. Its CSD/HSD are not the lowest, so higher mean synchrony does not imply lower variability

across clips. **(ii) Distributional audio quality.** Dyna2Music obtains the lowest FAD in all three feature spaces on both datasets, although the strongest competing method varies by feature extractor. This consistent ranking supports closer agreement with the reference music distributions across the three representations.



(a) AIST++ (b) TikTok

Figure 3: Music-to-kinematic BAS ($\sigma = 50$ ms; higher is better) using the five-part, two-band motion reference on (a) AIST++ and (b) TikTok. GT denotes paired music; colors identify method groups.

4.4 Beat Alignment Across Motion Granularities

We evaluate music-to-kinematic BAS across the five motion-reference configurations introduced in Figure 1. With the five-part, two-band reference (Figure 3), Dyna2Music achieves the highest score among generation methods on both datasets: 0.684 on AIST++ and 0.654 on TikTok, exceeding the strongest baseline by 0.023 in each case. Both scores remain below those of the paired GT music. It also ranks first under all four additional configurations in Appendix C (Figures A2 and A3). The consistent ranking indicates that its advantage persists across spatial partitions and frequency treatments of the evaluation reference. These comparisons vary the scoring reference while keeping each trained model fixed. Complementary ablations of the conditioning representation are reported in Appendix C.1.

4.5 Beat-Relative Rhythm Analysis

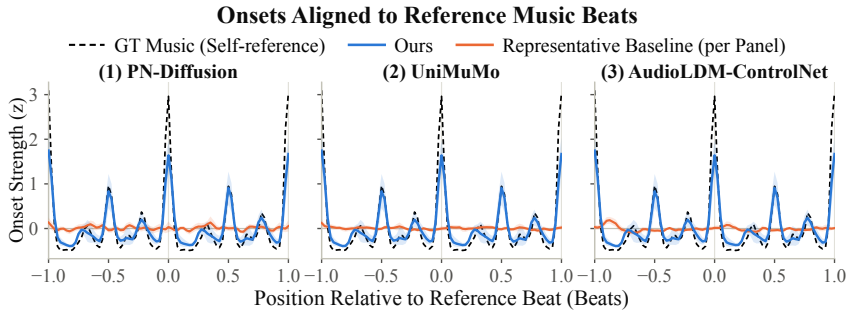
We examine onset structure relative to the paired music's beat grid on 180 AIST++ test clips spanning 10 tracks (Figure 4). Processing details and matched-minus-mismatched comparisons are provided in Appendix B.2 (Figure A1).

Dyna2Music shows more pronounced peaks at reference beats than the representative baselines, with additional smaller peaks at half-beat subdivisions. Its autocorrelation peaks recur at one- and two-beat lags. These patterns support agreement with the reference rhythm in both onset placement and periodicity. However, shifting motion conditions by approximately a quarter or half a beat leaves the onset peaks near their original phase. This observation is consistent with a clip-onset phase prior and limits claims of local motion-dependent phase tracking.

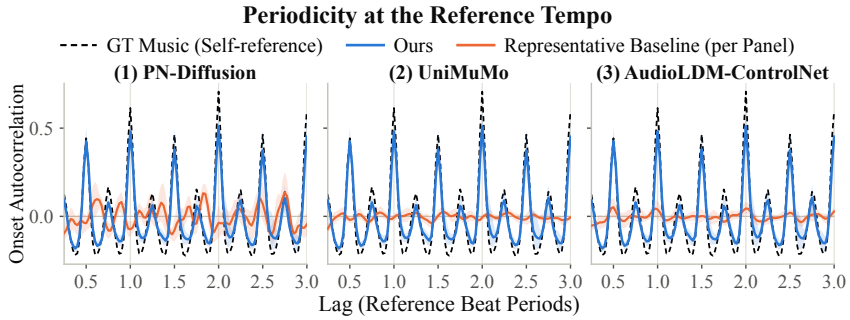
4.6 Output Diversity Across Dances

Figure 5 measures similarity between outputs for different dances using LAION-CLAP embeddings (Wu et al., 2023); pair construction and estimation details are given in Appendix B.3. Dyna2Music's mean similarity is 0.58 on AIST++ and 0.47 on TikTok, compared with 0.62 and 0.66 for GT music. The confidence intervals overlap on AIST++, while Dyna2Music's interval lies below GT on TikTok. In contrast, D2M-GAN and CMT have intervals entirely above GT on both benchmarks, indicating more homogeneous outputs in this feature space. Dyna2Music therefore exhibits greater variation across dances than these baselines.

Lower similarity alone does not establish musical quality or responsiveness to motion, since outputs can differ without matching their conditioning dance. The statistic characterizes variation across inputs; it does not measure repeated-sample diversity for a fixed dance or novelty relative to training music.

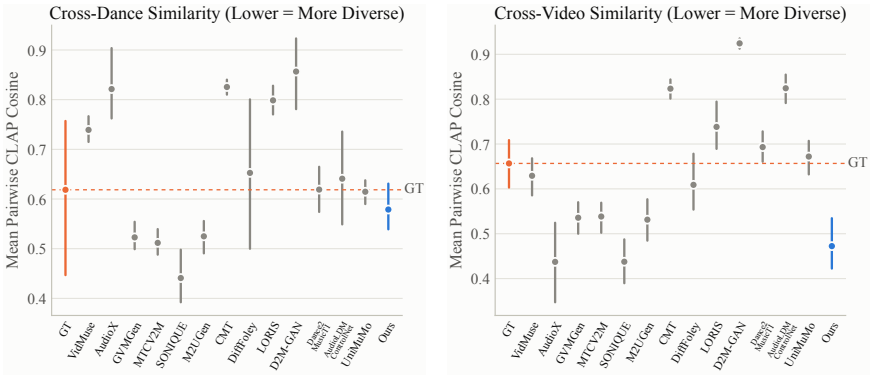


(a) Beat-relative onset profiles



(b) Onset autocorrelation

Figure 4: Beat-relative rhythm on AIST++: (a) standardized onset envelopes aligned to reference-music beats; (b) autocorrelation with lag normalized by the reference track's nominal beat period. Each column compares Dyna2Music with one baseline; GT is evaluated against its own timing. Shading denotes 95% bootstrap confidence intervals over 10 tracks.



(a) AIST++ (b) TikTok

Figure 5: Mean pairwise CLAP cosine similarity across different dances: (a) AIST++ (180 sequence pairs); (b) TikTok (1,358 clip pairs). The dashed line marks the corresponding similarity of GT music. Error bars show 95% bootstrap intervals over music tracks (AIST++) or source videos (TikTok). Higher values indicate more similar embeddings.

5 Conclusion

We presented **Dyna2Music**, a latent flow-matching framework for variable-length dance-to-music generation. It combines hierarchical fusion of part-specific slow-fast motion cues and pretrained joint features with latent dynamics consistency, which explicitly supervises local temporal variation in music latents. Experiments on AIST++ and TikTok show higher onset-level synchrony and lower FAD than the evaluated baselines, with consistent BAS advantages across motion-reference configurations. Future work will address finer motion-dependent phase control and long-range musical structure.

AI USE STATEMENT

In preparing this manuscript, we used generative AI tools solely for language refinement, including correcting grammatical and spelling errors and improving clarity and readability. Generative AI tools were not used for research conceptualization, methodological design, data analysis, or interpretation of results. The authors reviewed all revisions and take full responsibility for the final content of the manuscript.

REFERENCES

Gunjan Aggarwal and Devi Parikh. Dance2music: Automatic dance-driven music generation. *arXiv preprint arXiv:2107.06252*, 2021.

Sebastian Böck and Gerhard Widmer. Maximum filter vibrato suppression for onset detection. In *Proc. of the 16th Int. Conf. on Digital Audio Effects (DAFx). Maynooth, Ireland (Sept 2013)*, volume 7, pp. 4. Citeseer, 2013.

Shangzhe Di, Zeren Jiang, Si Liu, Zhaokai Wang, Leyan Zhu, Zexin He, Hongming Liu, and Shuicheng Yan. Video background music generation with controllable music transformer. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 2037–2045, 2021.

Simon Dixon. Onset detection revisited. In *Proceedings of the 9th international conference on digital audio effects*, volume 120, pp. 17. Citeseer Princeton, NJ, USA, 2006.

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap learning audio concepts from natural language supervision. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba. Foley music: Learning to generate music from videos. In *European Conference on Computer Vision*, pp. 758–775. Springer, 2020.

Bo Han, Yuheng Li, Yixuan Shen, Yi Ren, and Feilin Han. Dance2midi: Dance-driven multi-instrument music generation. *Computational Visual Media*, 10(4):791–802, 2024.

Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pp. 131–135. IEEE, 2017.

Shulei Ji, Songruoyao Wu, Zihao Wang, Shuyu Li, and Kejun Zhang. A comprehensive survey on generative ai for video-to-music generation. *arXiv preprint arXiv:2502.12489*, 2025.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

Ryota Kimura, Sangheon Park, Natalia Polouliakh, and Taketo Akama. Dance to music generation leveraging pre-training with unpaired data and contrastive alignment. *arXiv preprint arXiv:2607.10537*, 2026.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.

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

Sifei Li, Weiming Dong, Yuxin Zhang, Fan Tang, Chongyang Ma, Oliver Deussen, Tong-Yee Lee, and Changsheng Xu. Dance-to-music generation with encoder-based textual inversion. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024.

Xiao Liang, Wensheng Li, Lifeng Huang, and Chengying Gao. Dancecomposer: dance-to-music generation using a progressive conditional music generator. *IEEE Transactions on Multimedia*, 26: 10237–10250, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Shansong Liu, Atin Sakkeer Hussain, Qilong Wu, Chenshuo Sun, and Ying Shan. M^2UGen: Multimodal music understanding and generation with the power of large language models. *arXiv preprint arXiv:2311.11255*, 2023.

Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *Advances in Neural Information Processing Systems*, 36: 48855–48876, 2023.

Ziqian Ning, Huakang Chen, Yuepeng Jiang, Chunbo Hao, Guobin Ma, Shuai Wang, Jixun Yao, and Lei Xie. Diffrhythm: Blazingly fast and embarrassingly simple end-to-end full-length song generation with latent diffusion. *arXiv preprint arXiv:2503.01183*, 2025.

Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel PW Ellis, and C Colin Raffel. Mir_eval: A transparent implementation of common mir metrics. In *ISMIR*, volume 10, pp. 2014, 2014.

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*, pp. 11050–11059, 2022.

Robert C Streijl, Stefan Winkler, and David S Hands. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227, 2016.

Kun Su, Xiulong Liu, and Eli Shlizerman. How does it sound? *Advances in Neural Information Processing Systems*, 34:29258–29273, 2021.

Changchang Sun, Gaowen Liu, Charles Fleming, and Yan Yan. Enhancing dance-to-music generation via negative conditioning latent diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 8321–8330, 2025.

Vanessa Tan, Junghyun Nam, Juhan Nam, and Junyong Noh. Motion to dance music generation using latent diffusion model. In *SIGGRAPH Asia 2023 Technical Communications*, pp. 1–4. 2023.

Zeyue Tian, Zhaoyang Liu, Ruibin Yuan, Jiahao Pan, Qifeng Liu, Xu Tan, Qifeng Chen, Wei Xue, and Yike Guo. Vidmuse: A simple video-to-music generation framework with long-short-term modeling. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18782–18793. IEEE, 2025.

Zeyue Tian, Zhaoyang Liu, Yizhu Jin, Ruibin Yuan, Liumeng Xue, Xu Tan, Qifeng Chen, Wei Xue, and Yike Guo. Audiox: a unified framework for anything-to-audio generation. In *International Conference on Learning Representations*, volume 2026, pp. 110625–110651, 2026.

Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pp. 501–510, 2019.

Junxian Wu, Weitao You, Heda Zuo, Dengming Zhang, Pei Chen, and Lingyun Sun. Controllable video-to-music generation with multiple time-varying conditions. In *Proceedings of the 33rd ACM international conference on multimedia*, pp. 10427–10436, 2025.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Han Yang, Kun Su, Yutong Zhang, Jiaben Chen, Kaizhi Qian, Gaowen Liu, and Chuang Gan. Unimumo: Unified text, music, and motion generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25615–25623, 2025.

Jiashuo Yu, Yaohui Wang, Xinyuan Chen, Xiao Sun, and Yu Qiao. Long-term rhythmic video soundtracker. In *International Conference on Machine Learning*, pp. 40339–40353. PMLR, 2023.

Chaoyang Zhang and Yan Hua. Dance2music-diffusion: leveraging latent diffusion models for music generation from dance videos. *EURASIP Journal on Audio, Speech, and Music Processing*, 2024 (1):48, 2024.

Liqian Zhang and Magdalena Fuentes. Sonique: Video background music generation using unpaired audio-visual data. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025.

Wentao Zhu, Xiaoxuan Ma, Zhaoyang Liu, Libin Liu, Wayne Wu, and Yizhou Wang. Motionbert: A unified perspective on learning human motion representations. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15085–15099, 2023.

Ye Zhu, Kyle Olszewski, Yu Wu, Panos Achlioptas, Menglei Chai, Yan Yan, and Sergey Tulyakov. Quantized gan for complex music generation from dance videos. In *European Conference on Computer Vision*, pp. 182–199. Springer, 2022a.

Ye Zhu, Yu Wu, Kyle Olszewski, Jian Ren, Sergey Tulyakov, and Yan Yan. Discrete contrastive diffusion for cross-modal music and image generation. *arXiv preprint arXiv:2206.07771*, 2022b.

Heda Zuo, Weitao You, Junxian Wu, Shihong Ren, Pei Chen, Mingxu Zhou, Yujia Lu, and Lingyun Sun. Gvmgen: A general video-to-music generation model with hierarchical attentions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 23099–23107, 2025.

APPENDIX

Appendix A details the implementation in Section 3. Appendices B and C supplement the evaluation protocols and BAS comparisons in Section 4. The remaining appendices cover LDC-weight sensitivity, the user study, limitations, and broader impact.

A IMPLEMENTATION DETAILS OF DYNA2MUSIC

A.1 MOTION PREPROCESSING AND SSMD

This subsection details motion preprocessing for SSMD (Section 3.2).

Motion inputs and joint partition. Input poses follow the H36M-17 convention at the native video frame rate. Motion-energy computation and MotionBERT feature pooling use the same five-part partition, with zero-based joint indices:

Body part	Joint indices
Torso	$\{0, 7, 8, 9, 10\}$
Left arm	$\{11, 12, 13\}$
Right arm	$\{14, 15, 16\}$
Left leg	$\{4, 5, 6\}$
Right leg	$\{1, 2, 3\}$

The torso includes the root, neck, head, and other axial joints. Velocities are consecutive pose differences multiplied by the video frame rate, with the first velocity set to zero. The energy branch uses the supplied coordinates without per-frame torso centering.

Filtering and temporal alignment. We apply fourth-order Butterworth low-pass (1.4 Hz) and high-pass (3.0 Hz) filters forward and backward along each velocity coordinate. After computing mean squared speed within each part, we cap energy at the 99.5th percentile per clip, part, and band. Energy sequences are Fourier-resampled to the target latent length and clamped to nonnegative values. Training uses the paired music-latent length; pose-only inference uses $\lfloor (N/f_v)(44100/2048) \rfloor$, subject to the length cap in Appendix A.3, where f_v is the video frame rate.

Frozen MotionBERT extracts 512-dimensional features for all 17 joints in windows of up to 243 frames. Features for a final segment shorter than 16 frames are computed in an overlapping window ending at the sequence boundary and cropped to that segment. Part-wise averaging and Fourier resampling yield joint features of shape $T \times 5 \times 512$, aligned with the slow and fast energy sequences of shape $T \times 5$. Percentile capping and nonnegative clamping apply only to energy.

A.2 HSF-ENCODER ARCHITECTURE

This subsection specifies the HSF-Encoder in Section 3.3.

Architecture. Layer-normalized joint features are projected to 256 dimensions; scalar energy inputs use two-layer MLPs with hidden width 128 and SiLU activation. Energy is standardized over valid time steps per sample, part, and band, with standard deviations lower-bounded by 10^{-5}. Four-head attention uses band-specific query projections and shared key, value, and output projections. Each band has a $256 \to 128 \to 1$ part-gating MLP with SiLU and a softmax over parts. The fusion gate maps the normalized 512-dimensional branch concatenation to a scalar sigmoid output. The final projection combines that concatenation with the 256-dimensional gated mixture, mapping 768 dimensions to 256.

HSF embeddings and attention. Let $\mathbf{B}^s = \mathbf{S}$ and $\mathbf{B}^f = \mathbf{F}$, and let $\mathcal{Z}$ denote the masked temporal standardization described above. The embeddings used in Section 3.3 are

$$\begin{aligned} \widetilde{\mathbf{E}}_{n,p} &= \mathbf{W}_E \mathrm{LN}(\mathbf{E}_{n,p}) + \mathbf{e}_n^{\mathrm{time}} + \mathbf{e}_p^{\mathrm{part}}, \\ \widetilde{\mathbf{B}}^b_{n,p} &= \mathrm{MLP}_b\big(\mathcal{Z}(\mathbf{B}^b)_{n,p}\big) + \mathbf{e}_n^{\mathrm{time}} + \mathbf{e}_p^{\mathrm{part}} + \mathbf{e}_b^{\mathrm{band}}. \end{aligned} \tag{A1}$$

Temporal embeddings are sinusoidal; part and band embeddings are learned. For band b and head h, queries $\mathbf{Q}^b_{n,h}$ are projected from $\widetilde{\mathbf{B}}^b_n$, while keys $\mathbf{K}^{\mathrm{attn}}_{n,h}$ and values $\mathbf{V}_{n,h}$ are projected from $\widetilde{\mathbf{E}}_n$. The operator MHA_b in Eq. 3 expands as

$$\mathbf{A}^b_{n,h} = \mathrm{softmax}\left(\frac{\mathbf{Q}^b_{n,h}(\mathbf{K}^{\mathrm{attn}}_{n,h})^{\mathsf{T}}}{\sqrt{d_h}}\right), \qquad \mathbf{H}^b_n = \mathrm{Concat}_h\big(\mathbf{A}^b_{n,h}\mathbf{V}_{n,h}\big)\,\mathbf{W}_O. \tag{A2}$$

With four heads, $d_h = d/4$. Each row of $\mathbf{A}^b_{n,h} \in \mathbb{R}^{5\times 5}$ is normalized over the five key parts at the same time step. Padded conditioning tokens are zeroed and excluded from the music generator's cross-attention.

A.3 Music Backbone and Sequence Lengths

This subsection details the generator in Section 3.4 and sequence lengths used in Section 4.2.

Music backbone. The transformer has 16 blocks, hidden width 2048, 32 attention heads, and feed-forward expansion factor four. Each block applies noncausal self-attention with rotary positional embeddings and a SiLU-gated feed-forward network, followed by separately normalized residual cross-attention to the HSF tokens. The input concatenates 64 music-latent channels with a 512-dimensional flow-time embedding before projection to the hidden width. The positional module uses two temporal convolutions with kernel size 31, 16 groups, and Mish activations. Final adaptive normalization predicts scale and shift from the flow-time embedding before projection to 64 output channels.

Waveform representation and sequence lengths. The frozen DiffRhythm VAE (Ning et al., 2025) uses 64 latent channels, temporal downsampling by 2048, and stereo 44.1 kHz output. Training reads precomputed reference latents; inference decodes generated latents to waveforms. The loader requires at least 100 latent frames (approximately 4.6 s), excluding five of the 391 TikTok training pairs and leaving 386 eligible pairs. All AIST++ pairs and all 53 TikTok test pairs meet this threshold. Training sequences are padded to the configured caps: 1200 latent frames for AIST++ and 700 for TikTok (approximately 55.7 s and 32.5 s). Valid lengths determine attention masks, energy-normalization statistics, and the averaging sets in Eqs. A3 and 9. Inference supports variable durations up to a configurable cap, which defaults to 1200 latent frames; longer inputs are truncated.

A.4 Training Objective and Sampling

This subsection complements Section 3.4; optimization settings appear in Section 4.2.

Flow-time sampling and velocity regression. For each sample q, we independently draw $t_q = \sigma(\xi_q)$ with $\xi_q \sim \mathcal{N}(0,1)$. Let $\mathcal{I}$ contain the valid sample-frame pairs (q,n). The flow-matching loss averages squared velocity error over these pairs and the d_a latent channels:

$$\mathcal{L}_{\mathrm{FM}} = \frac{1}{d_a|\mathcal{I}|}\sum_{(q,n)\in\mathcal{I}} \big\|\mathbf{v}_\theta(\mathbf{x}_{t_q,q}, t_q, \mathbf{C}_q)_n - (\mathbf{x}_{1,q,n} - \mathbf{x}_{0,q,n})\big\|_2^2. \tag{A3}$$

With probability 0.2, a shared minibatch-level decision zeros all HSF tokens to enable classifier-free guidance.

LDC computation. The single-step estimate in Eq. 7 is differentiable and recovers the paired endpoint when the predicted velocity equals $\mathbf{v}^\star$. LDC reuses the flow-matching velocity prediction and is evaluated in float32 over adjacent valid frames, with the reference latent detached from the gradient graph. Pairs involving padding are excluded. Computation remains in latent space and requires neither beat annotations nor waveform decoding.

Generation. Using exponential-moving-average weights, Euler integration maps $\mathbf{x}(0) \sim \mathcal{N}(0,\mathbf{I})$ to the latent endpoint:

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{v}^{\mathrm{cfg}}_\theta(\mathbf{x}(t), t, \mathbf{C}), \qquad \widehat{\mathbf{y}} = \mathcal{D}(\mathbf{x}(1)), \tag{A4}$$

where optional classifier-free guidance is implemented as

$$\mathbf{v}^{\mathrm{cfg}}_\theta = \mathbf{v}_\theta(\mathbf{x}, t, \mathbf{C}) + \gamma[\mathbf{v}_\theta(\mathbf{x}, t, \mathbf{C}) - \mathbf{v}_\theta(\mathbf{x}, t, \mathbf{0})]\,. \tag{A5}$$

The generation scripts use $\gamma = 0$, corresponding to the conditional velocity without guidance amplification. Sampling uses 32 uniformly spaced time points $t_k = k/31$ ($k = 0, \ldots, 31$), giving 31 Euler updates:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + (t_{k+1} - t_k)\mathbf{v}_\theta^{\mathrm{cfg}}(\mathbf{x}^{(k)}, t_k, \mathbf{C}). \tag{A6}$$

The frozen VAE decoder converts the endpoint to stereo audio; LDC and the VAE encoder are not evaluated during sampling.

B Evaluation Details

B.1 Onset-Level Synchrony

This protocol specifies the onset-level metrics in Table 1 (Section 4.2).

Relation to prior evaluation. LORIS (Yu et al., 2023) computes Beats Coverage Score (BCS) and Beats Hit Score (BHS) from one-second bins of detected onsets. Binning merges events and discards within-bin timing, potentially obscuring temporal mismatches. We retain the precision/recall interpretations but match continuous onset timestamps within ± 50 ms.

Detection and matching. Complete audio signals are resampled to 44.1 kHz. Generated and reference music use identical onset detectors: a SuperFlux-style spectral-flux envelope (Böck & Widmer, 2013), median aggregation, and a 256-sample hop (5.8 ms). Maximum-cardinality bipartite matching in *mir_eval* (Raffel et al., 2014) pairs onsets one-to-one within ± 50 ms (Dixon, 2006). Scoring covers the full reference duration: excess generated events are discarded, and uncovered intervals in shorter outputs contain no generated events.

Metrics and interpretation. For each clip, BCS and BHS equal the matched count divided by the generated and reference onset counts, respectively; F1 is their harmonic mean. Table 1 reports clip-averaged BCS, BHS, and F1, and the sample standard deviations of BCS and BHS (CSD and HSD). All five statistics are expressed as percentages. Because chance matches depend on onset density, these scores quantify event synchrony with a particular recording and alone do not establish dance-specific rhythmic alignment. The complementary beat-relative analysis in Section 4.5 uses the mismatched-track controls detailed below.

B.2 Beat-Relative Onset Structure and Periodicity

Protocol. Section 4.5 analyzes 234 five-second windows from 180 AIST++ test clips spanning 10 music tracks. Audio is resampled to 22.05 kHz; onset-strength envelopes use a 256-sample hop and are standardized within each window. Beat grids are estimated from complete reference recordings, with nominal tempo initializing the beat tracker. Intervals before and after each reference beat are normalized separately to $[-1, 0]$ and $[0, 1]$ beats before averaging aligned envelopes (Figure 4(a)). Autocorrelation lags are normalized by the reference track's nominal beat period (Figure 4(b)). Generated audio supplies the onset envelopes; no generated beat sequence is required.

Mismatched controls and estimation. Each generated window is also paired with 20 randomly sampled beat grids from other tracks, whose nominal beat periods define the periodicity controls. The on-beat statistic averages onset strength within ± 0.1 beats; the periodicity statistic averages autocorrelation within ± 0.05 beats of one- and two-beat lags. Figure A1 reports matched-minus-mismatched differences. We average windows within clips, clips within tracks, and then tracks with equal weight. The 95% confidence intervals use the 2.5th and 97.5th percentiles of $10{,}000$ track-bootstrap resamples (seed 20260921). GT uses its own timing as a self-reference.

Results. Dyna2Music has the largest contrasts among generation methods for both on-beat onset strength (0.379; 95% CI $[0.256, 0.487]$) and periodicity (0.272; $[0.232, 0.315]$). AudioLDM-ControlNet shows a smaller positive on-beat contrast (0.035; $[0.012, 0.062]$), while most baselines' intervals include zero. CMT has positive periodicity contrast (0.108; $[0.058, 0.160]$) but an on-beat interval containing zero, illustrating that periodicity and phase agreement provide distinct evidence. PN-Diffusion has negative contrasts on both measures (-0.077 and -0.095). These comparisons support reference-rhythm correspondence; local motion-dependent phase tracking remains limited, as discussed in Section 4.5.

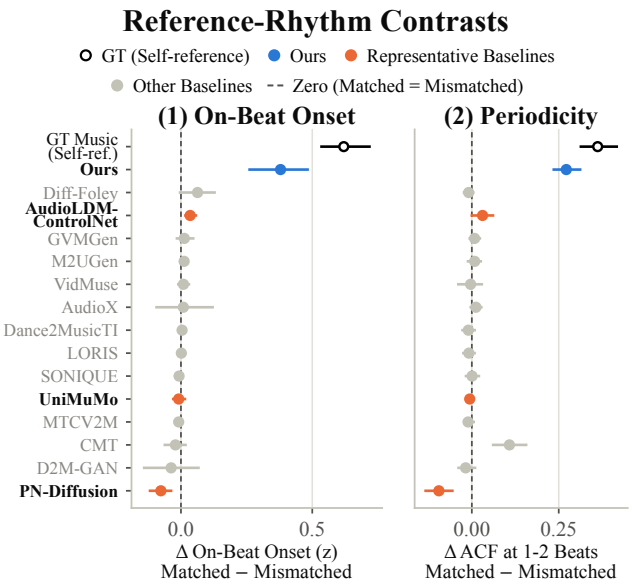


Figure A1: Matched-minus-mismatched rhythm statistics on AIST++: (1) onset strength within ± 0.1 beats of reference beats; (2) autocorrelation within ± 0.05 beats of one- and two-beat lags. Points and bars show track-averaged differences and 95% bootstrap confidence intervals; zero denotes equal matched and mismatched scores.

B.3 Cross-Dance Output Similarity

The analysis in Section 4.6 and Figure 5 uses clip representations obtained by averaging normalized LAION-CLAP embeddings from five-second windows with a one-second hop. Cosine similarity between clip representations measures output similarity. AIST++ pairs use sequences with different music tracks and the same camera view; averaging over nine views yields 180 sequence pairs from 20 sequences and 10 tracks. For TikTok, excluding same-source pairs among 53 clips from 43 videos leaves 1,358 pairs.

We compute the mean over eligible pairs for each method and GT music. The 95% intervals use 1,000 cluster-bootstrap resamples over music tracks (AIST++) or source videos (TikTok). Each pair receives the product of its two clusters' resampling counts, accounting for dependence from shared tracks or videos. The statistic describes variation across inputs, rather than repeated-sample diversity for a fixed dance or novelty relative to training music.

C Motion Granularity Analyses

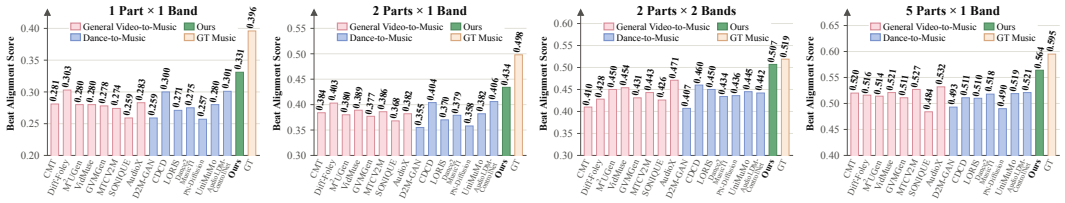
Figures A2 and A3 supplement the five-part, two-band comparison in Section 4.4 with four additional motion references.

Motion references. One part uses the full skeleton; two parts group the torso and arms as the upper body and both legs as the lower body; five parts follow Appendix A.1. One band uses unfiltered velocity; two bands use the 1.4 Hz low-pass and 3.0 Hz high-pass components. For each part and band, kinematic beats are local minima of mean joint speed after Gaussian smoothing (2.5-frame standard deviation). Minima require prominence of 0.3 times the curve's standard deviation and separation of at least 0.1 s; their union forms the motion reference. Evaluation therefore uses speed minima as event times, whereas SSMD uses mean squared speed as conditioning energy (Eq. 2).

Music-to-kinematic scoring. Music beats are detected from audio resampled to 44.1 kHz using a 512-sample hop, and scoring uses the common duration of the audio and motion sequences. Given detected music beats $\mathcal{M}$ and kinematic beats $\mathcal{K}$, the score is

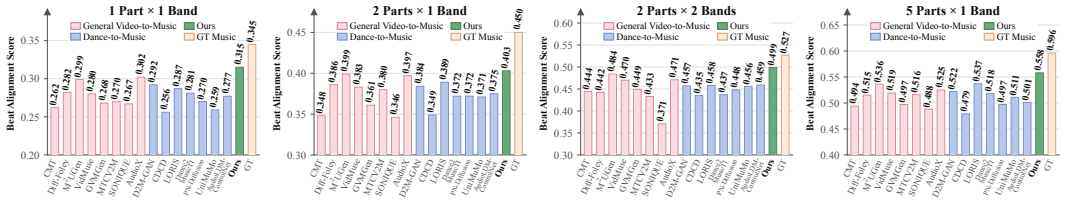
$$\mathrm{BAS}(\mathcal{M},\mathcal{K}) = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \exp\left(-\frac{\min_{k\in\mathcal{K}}(m-k)^2}{2\sigma^2}\right), \qquad \sigma = 0.05\,\mathrm{s}. \tag{A7}$$

All methods use the same reference within each configuration; GT denotes paired music scored against that reference. Dyna2Music ranks first among generation methods in all four additional configurations on both datasets, consistent with Section 4.4. These comparisons vary the scoring reference while keeping trained models fixed. Raw BAS depends on reference density, so absolute scores across configurations require care in interpretation.



(a) AIST++ 1 part × 1 band (b) AIST++ 2 parts × 1 band (c) AIST++ 2 parts × 2 bands (d) AIST++ 5 parts × 1 band

Figure A2: Music-to-kinematic BAS ($\sigma = 50$ ms) on AIST++ under four additional motion references. Panel labels specify part and band counts; colors follow Figure 3. Higher values indicate closer alignment within each configuration.



(a) TikTok 1 part × 1 band (b) TikTok 2 parts × 1 band (c) TikTok 2 parts × 2 bands (d) TikTok 5 parts × 1 band

Figure A3: Music-to-kinematic BAS ($\sigma = 50$ ms) on TikTok under four additional motion references. Panel labels specify part and band counts; colors follow Figure 3. Higher values indicate closer alignment within each configuration.

C.1 Ablation of Motion Conditioning

We compare separately trained conditioning variants using the same onset-level evaluation protocol as Table 1. One-, two-, and five-part variants use the whole body, upper/lower body, and the five-part partition in Appendix A.1, respectively. For the one-band variants, slow and fast energies are summed and supplied to both encoder branches; two-band variants retain separate signals. Part merging averages both energy and joint features with weights proportional to joint counts. Thus, these experiments alter the model's conditioning inputs, complementing the fixed-model BAS reference comparisons above.

Table A1: Ablation of motion conditioning on AIST++ and TikTok using onset-level metrics (%). One band denotes merged slow/fast energy. Full-model results are reproduced from Table 1; best values within each dataset are bold.

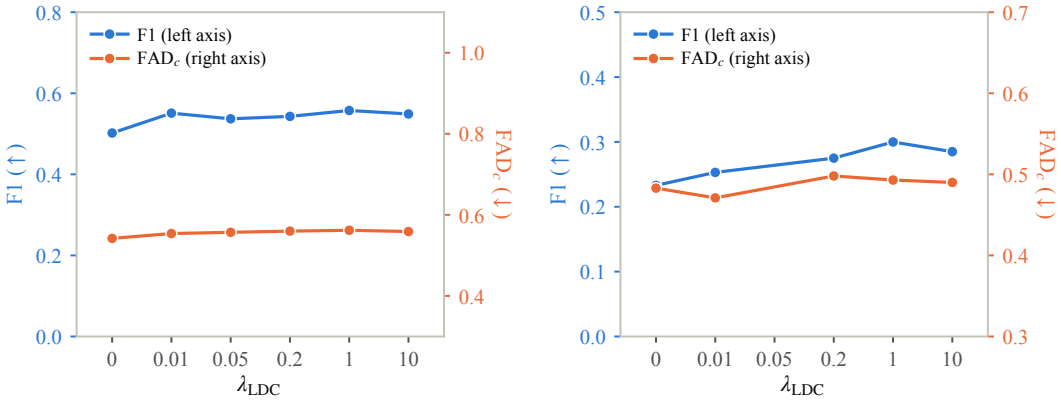
Dataset	Parts × Bands	BCS↑	CSD↓	BHS↑	HSD↓	F1↑
AIST++	1 × 1	53.20	26.31	52.87	25.48	47.49
	2 × 1	47.14	24.98	58.43	**22.95**	47.66
	2 × 2	45.92	**24.30**	52.34	24.31	43.58
	5 × 1	50.43	26.10	56.09	26.18	47.38
	5 × 2 (full)	**60.44**	26.57	**63.54**	25.70	**55.76**
TikTok	1 × 1	19.71	**11.29**	26.66	**11.79**	20.89
	2 × 1	19.12	11.53	25.10	12.45	20.23
	2 × 2	18.11	**11.29**	26.55	14.38	20.01
	5 × 1	21.13	14.00	24.77	12.25	20.45
	5 × 2 (full)	**24.06**	11.44	**44.19**	15.08	**29.99**

Table A1 shows that the full five-part, two-band model achieves the highest BCS, BHS, and F1 on both datasets. Merging the frequency bands at five-part resolution reduces F1 by 8.38 percentage points on AIST++ and 9.54 on TikTok. Retaining both bands but merging the five parts into upper/lower body reduces F1 by 12.18 and 9.98 points, respectively. These results favor retaining both part and frequency distinctions in the reported models. However, band separation at two-part resolution does not improve F1, and finer partitioning alone gives no consistent gains among the one-band variants. The benefit therefore depends on the combined representation; differing training budgets limit attribution to conditioning structure alone.

D Sensitivity to the LDC Weight

Figure A4 examines the LDC weight in Eq. 9 using onset-level F1 (the ± 50 ms metric in Table 1) and CLAP-based FAD_c. We evaluate $\{0, 0.01, 0.05, 0.2, 1, 10\}$ on AIST++ and $\{0, 0.01, 0.2, 1, 10\}$ on TikTok; zero recovers flow matching alone.

All evaluated nonzero weights improve F1 over zero on both datasets. F1 peaks at $\lambda_{\text{LDC}} = 1$ (0.5576 on AIST++ and 0.2999 on TikTok), compared with 0.502 and 0.233 at zero, and declines to 0.549 and 0.285 at weight 10. The lowest FAD_c instead occurs at zero on AIST++ (0.542) and 0.01 on TikTok (0.471), compared with 0.562 and 0.493 at weight 1. Thus, stronger LDC supervision does not monotonically improve synchrony, and the weight with the highest F1 differs from that with the lowest FAD_c among the evaluated settings.



(a) AIST++ (b) TikTok

Figure A4: Sensitivity to λ_{LDC} on (a) AIST++ and (b) TikTok. Blue: onset-level F1 on a 0-1 scale (left; higher is better). Orange: FAD_c (right; lower is better). Horizontal positions are categorical; markers denote evaluated weights, with no 0.05 evaluation on TikTok.

E Subjective Evaluation

We compare Dyna2Music with AudioLDM-ControlNet and SONIQUE on AIST++, selecting these representative baselines for their generation quality (Section 4.2). The study involves 10 participants, each completing 10 evaluation trials. Each trial presents three versions of the same dance video, paired with soundtracks generated by the three methods. Participants rate each version separately for overall musical quality (OVL) and dance-music relevance (REL) on a five-point scale, following Sun et al. (2025), yielding 100 ratings per method for each dimension. REL assesses perceived rhythmic correspondence with the dance. Presentation order is randomized and method identities are concealed. We report mean opinion scores (MOS) (Streijl et al., 2016) for each dimension, with higher values indicating better perceived quality or relevance.

Table A2 reports mean OVL and REL scores of 4.3 and 4.5 for Dyna2Music, respectively. These exceed the highest baseline scores by 1.0 point for OVL (AudioLDM-ControlNet) and 1.1 points for REL (SONIQUE). The ratings support improvements in both perceived musical quality and dance-music relevance relative to the selected baselines.

Table A2: Subjective evaluation on AIST++: mean opinion scores for overall musical quality (OVL) and dance-music relevance (REL) on a 1-5 scale (higher is better). Best scores are bold.

Method	OVL↑	REL↑
AudioLDM-ControlNet	3.3	3.0
SONIQUE	3.1	3.4
Dyna2Music (Ours)	**4.3**	**4.5**

F LIMITATIONS

Motion quality and temporal control. Pose-estimation errors from occlusion or motion blur can affect both energy and joint features. The conditioning-shift checks in Section 4.5 indicate that reference-beat agreement does not establish local phase tracking of motion accents.

Data and duration. AIST++ and TikTok cover studio and in-the-wild settings but a limited range of dance styles, music, and recording conditions. Variable-length generation remains subject to the configured cap (Appendix A.3).

G BROADER IMPACT

Dance-to-music generation may support video production, creative exploration, and choreography practice. Potential concerns include effects on professional music work and resemblance to existing recordings or artists' styles. Use should disclose machine generation and respect relevant rights and licensing conditions. Our evaluations do not establish the originality or licensing status of individual outputs.