HEDGE: HIERARCHICAL EXPRESSIVE DANCE GENERATION WITH LONG-TERM MUSIC STRUCTURE GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating dance from music not only requires precise beat alignment but also synchronization with the global musical structure, especially in long sequences containing multiple sections such as intros, verses, and choruses. Existing methods primarily focus on beat-level alignment, but often fail to capture and reflect broader musical dynamics, which limits expressive variation and results in choreography that does not adapt well to section-level transitions. To address these limitations, we introduce hierarchical segmentation for both music and motion, organizing them into section and interval levels to more effectively capture structural and rhythmic details. Building on this representation, we propose HEDGE, a two-level diffusion-based framework that generates dance sequences conditioned on both local rhythmic cues and global musical structure. Our method produces choreography that is not only realistic and musically synchronized, but also dynamically adapts to the expressive changes across musical sections. Extensive experiments demonstrate that HEDGE achieves superior responsiveness to musical structure, offering a significant advancement toward automated choreography that faithfully reflects the full narrative and expressive intent of music.

1 INTRODUCTION

Dance is a fundamental performing art with profound cultural significance, serving as a medium for storytelling, emotional expression, and social interaction. Historically, music and dance have been intrinsically linked, with musical elements such as rhythm, tempo, and sectional structure directly informing choreographic decisions and amplifying the emotional impact of performances. In longer musical works, distinct music sections, such as *"intro"*, *"verse"*, and *"chorus"*, guide choreographic choices to shape the movement flow and dance dynamics throughout extended pieces.

Recently, music-conditioned dance generation has attracted growing research interest due to its potential applications in automated choreography, virtual character animation, and interactive entertainment. The objective is to produce motion sequences that are both realistic and synchronized with musical beats, while also adapting to tempo changes and accurately reflecting different musical sections, such as the *"intro"* and *"chorus"*. While recent approaches have made significant progress in generating visually plausible dance movements synchronized with musical beats, they often struggle to capture the full temporal and structural dynamics of music. In particular, most existing methods rely on frame-level or short-window processing, which limits the model's ability to capture the broader musical narrative and can lead to the loss of important sectional information. Furthermore, the lack of global analysis across the entire sequence prevents the generated dances from aligning effectively with transitions between musical sections. Consequently, resulting dance often exhibits a uniform motion tempo and fails to reflect section-level dynamic shifts, for example, a gentle *"intro"* to an energetic *"chorus"*. This diminishes alignment between music and dance, highlighting the need for methods that deeply incorporate musical structure into dance generation.

To address these challenges, we propose a hierarchical representation for both music and motion. At the section-level, music and motion are organized according to the underlying musical structure, with section boundaries and labels automatically identified by LinkSeg Buisson et al. (2024). At the interval-level, data is segmented based on inter-beat intervals to capture finer rhythmic details. We

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.

leverage ACTOR-PAE from Au et al. (2025) to encode interval-level motion segments as phase latents, enabling compact and efficient summarization for section-level diffusion. These phase latents can be reparameterized into periodic latents, providing expressive and detailed representations for interval-level motion generation. Building on this hierarchical representation, we present HEDGE, a two-level diffusion framework for denoising dance sequences at both section and interval levels. At the section-level, phase latent sequences are denoised by incorporating section-level music features and contextual information from adjacent sections, ensuring coherent long-term alignment with musical structure. At the interval-level, periodic latent signals are refined through conditioning on the section-level phase latent, neighboring periodic latents, and detailed music features via specialized ControlNets, producing expressive and musically synchronized dance sequences. To demonstrate the effectiveness of our framework in capturing and responding to musical structure, we design evaluation metrics, inspired by the Jo-Ha-Kyu metrics in JoruriPuppet Dong et al. (2025), that assess motion speed and music tempo synchronization across different music sections. This enables us to quantitatively evaluate the responsiveness of generated dance sequences to the underlying musical structure. In summary, our contributions are as follows:

- We propose a hierarchical music and motion representation by segmenting data at both section and interval levels, using LinkSeg for automatic section identification and ACTOR-PAE to generate motion phase and periodic latents for two-level dance generation.
- We develop HEDGE, a two-level diffusion framework that denoises and generates dance sequences, ensuring musical structure coherence at the section-level and smooth, musically-aligned choreography at the interval-level.
- We validate our method through extensive experiments, employing metrics that assess motion speed and music tempo synchronization across music sections to evaluate how the generated dance responds to musical structure.

2 Related Works

2.1 Music-to-dance Generation

Music-to-motion generation has been widely explored to improve synchronization between music and dance. Early methods, such as motion graph-based methods Kovar et al. (2008); Au et al. (2022), autoregressively retrieve fixed-length dance segments in response to music cues. However, these retrieval-based methods require substantial storage and offer limited dance generation diversity.

Recently, deep learning-based methods have demonstrated strong potential for end-to-end music-to-motion generation. FACT Li et al. (2021) employs a Transformer Vaswani et al. (2017) to generate motion frame-by-frame, conditioned on input music and previous dance segments. Transflower Valle-Perez et al. (2021) extends this architecture by passing the Transformer output through a Normalizing Flow model Rezende & Mohamed (2015). Bailando Siyao et al. (2022), inspired by GPT-like Radford et al. (2019) models, encodes motion into a quantized codebook using VQ-VAE Van Den Oord et al. (2017) and autoregressively generates codes with a Transformer, while actor-critic Konda & Tsitsiklis (1999) is applied to improve beat alignment and body consistency.

Integration with diffusion models Ho et al. (2020); Song et al. (2020) has further enhanced motion generation quality. EDGE Tseng et al. (2023) autoregressively refines fixed-length dance segments, improving both motion quality and temporal continuity. LODGE Li et al. (2024) employs a two-level pipeline for long sequence generation, enabling the incorporation of choreography rules during dance primitive generation to enrich dance structure and music synchronization.

Despite these advances, most prior models rely on fixed-length and short windowed segments, which can disrupt sequence semantics and limit alignment with the underlying music. In contrast, our method segments dance sequences at variable lengths across two hierarchical levels, capturing both high-level musical structure and detailed rhythmic information for structure-aware dance generation.

2.2 Long Motion Sequence Generation

Long text-to-motion sequence generation closely resembles long dance generation, where variable-length motion segments are used to better align with text semantics. Existing frameworks adopt ei-

ther autoregressive or parallel generation strategies. In the autoregressive approach, TEACH Athanasiou et al. (2022) employs a variational encoder-decoder to autoregressively generate motion segments, each conditioned on individual text semantics. PCMDM Yang et al. (2023) leverages the Human Motion Diffusion Model Tevet et al. (2023) and introduces compositional transition sampling to improve the smoothness between segments. For parallel generation, PriorMDM Shafir et al. (2024) independently generates semantic motion segments in the first level, followed by the synthesis of blending transitional segments in the second level. TransPhase Au et al. (2025) refines this process by mutually denoising motion segments using both semantic and adjacent segment information. Building upon TransPhase, our pipeline segments long music into variable-length subsegments at both the section and interval levels, and employs a two-level diffusion process to improve structural coherence and music alignment in dance generation.

3 PRELIMINARIES

3.1 MUSIC STRUCTURE ANALYSIS

Music typically contains sections with varying tempos according to their structural labels. For example, *"chorus"* often has a faster tempo than *"intro"*. Given a long music track $\mathbf{m}$, we use LinkSeg Buisson et al. (2024) to automatically extract section labels $\mathbf{c}_s$, segmenting the music timeline based on seven categories: *"intro"*, *"verse"*, *"chorus"*, *"bridge"*, *"instrumental"*, *"outro"*, and *"silence"*. This process produces a variable number of sections, $\{\mathbf{m}_s\}_{s=1}^{S}$, where both the number of sections S and the length of each section $\mathbf{m}_s$ are adaptively determined by the structure of the input music track $\mathbf{m}$. While this adaptive segmentation effectively preserves structural information for music-motion alignment, the duration of each section $\mathbf{m}_s$ ranges from a few seconds to over a minute, often significantly longer than the segment lengths typically handled by existing motion diffusion pipelines.

3.2 PHASE-BASED MOTION ENCODING WITH ACTOR-PAE

A recent line of work models motion in a periodic motion latent space Starke et al. (2022); Au et al. (2025). In particular, Au et al. (2026) employs the ACTOR-PAE encoder $\mathcal{P}_E$ to transform an input motion sequence $\mathbf{x} \in \mathbb{R}^{T \times E_x}$ into four phase latent variables $\mathbf{F}, \mathbf{A}, \mathbf{B}, \mathbf{S} \in \mathbb{R}^{E_p}$, which are reparameterized as a periodic signal $\mathbf{q} \in \mathbb{R}^{T \times E_p}$:

$$\mathbf{q} = \mathbf{A} \sin(\mathbf{F} \cdot (N - \mathbf{S})) + \mathbf{B}, \tag{1}$$

where $N \in \mathbb{R}^T$ represents the temporal progression of the motion sequence. The ACTOR-PAE decoder $\mathcal{P}_D$ then reconstructs the motion $\hat{\mathbf{x}}$ from $\mathbf{p}$.

This periodic latent space $\mathbf{q}$ serves as the domain for diffusion-based motion generation and gradient-based refinement. As shown in Au et al. (2025; 2026), this representation enhances the connection between adjacent motion segments and regularizes motion throughout iterative optimization. Additionally, the phase latents $\mathbf{P} = [\mathbf{F}, \mathbf{A}, \mathbf{B}, \mathbf{S}]$ compactly encode each motion segment with four tokens, reducing tensor size and enabling efficient global profiling aligned with music structure.

4 HIERARCHICAL REPRESENTATION OF MUSIC AND MOTION

We propose to organize music data in two hierarchical levels: **Section-level**: Comprises a section of the music track associated with a specific structural label, such as *"chorus"*. **Interval-level**: Consists of music segments corresponding to inter-beat intervals.

To implement this hierarchical structure, we first segment the long music track $\mathbf{m} \in \mathbb{R}^{T \times E_a}$ into variable-length sections $\mathbf{m}_s \in \mathbb{R}^{T_s \times E_a}$ using the music structure analysis method from LinkSeg Buisson et al. (2024) (refer to Sec. 3.1), where each section is annotated with a structural label $\mathbf{c}_s$. For each section $\mathbf{m}_s$, we apply Librosa's beat tracking algorithm McFee et al. (2015) to extract beat onsets, which define the inter-beat intervals—the durations between two consecutive beats. Following the hierarchical structure shown in Fig. 1, we segment the long dance sequence at each level to learn and organize dance movements according to the underlying musical structure.

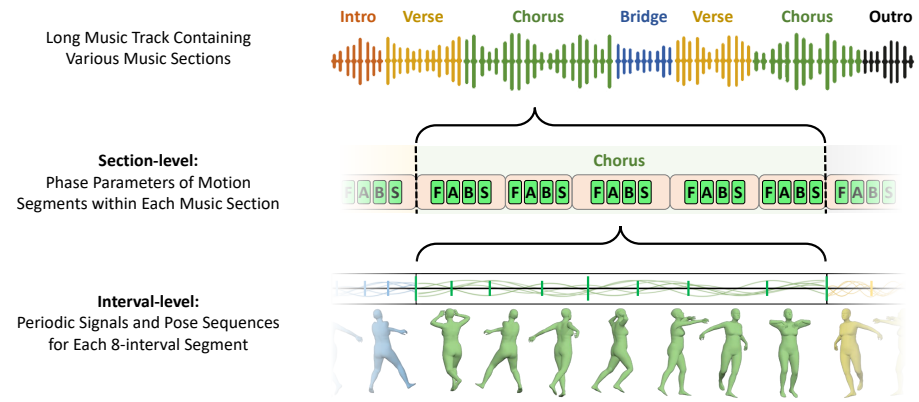


Figure 1: Hierarchical organization of music and motion data. The music track is segmented into two levels: section-level (structurally labeled segments) and interval-level (inter-beat intervals). Both music and motion sequences are segmented at each level. Section-level representations are obtained by aggregating interval-level segments into compact forms to enable efficient section-level learning.

At the interval-level, we organize the length of inter-beat intervals, typically resulting in intervals of 9 to 24 frames. Both the music track and corresponding motion are segmented into units of 8 inter-beat intervals with a 4-interval overlap, following the transitional motion setting in Au et al. (2025). We obtain interval-level motion segment $\mathbf{x}_i \in \mathbb{R}^{T_i \times E_x}$ and the corresponding music segment $\mathbf{m}_i \in \mathbb{R}^{T_i \times E_m}$, expressed in frame by frame. Each motion segment $\mathbf{x}_i$ is encoded using ACTOR-PAE, producing periodic latents $\mathbf{q}_i \in \mathbb{R}^{T_i \times E_p}$ as the interval-level representation.

At the section-level, the music section $\mathbf{m}_s$ and its corresponding motion section $\mathbf{x}_s$ are typically too long for conventional diffusion models, making training and inference inefficient. To address this, we aggregate information into more compact representations to facilitate learning. Specifically, each interval-level motion segment $\mathbf{x}_i$ within $\mathbf{x}_s$ is abstracted into its phase latents $\mathbf{P}_i = [\mathbf{F}, \mathbf{A}, \mathbf{B}, \mathbf{S}]$, so that each motion section $\mathbf{x}_s$ can be efficiently represented as a sequence of phase latents $\mathbf{P}_s = \{\mathbf{P}_i\}$ that collectively describe all interval-level motion segments within $\mathbf{x}_s$.

Similarly, each interval-level music segment $\mathbf{m}_i$ within $\mathbf{m}_s$ is abstracted into an aggregated section-level representation by computing and concatenating four types of weighted averages: $\mathbf{M}_i = [\textit{left}(\mathbf{m}_i), \textit{right}(\mathbf{m}_i), \textit{center}(\mathbf{m}_i), \textit{mean}(\mathbf{m}_i)]$, across the frames in each music segment. For example, the *left* weighted average assigns linearly decreasing weights from 1 to 0 across the frames in the segment $\mathbf{m}_i$. The details of the weighted averaging are provided in the Supplementary Material. The entire music section $\mathbf{m}_s$ can then be efficiently represented as a sequence $\mathbf{M}_s = \{\mathbf{M}_i\}$, collectively describing all interval-level music segments within $\mathbf{m}_s$.

5 The HEDGE Framework

We propose HEDGE, a two-level diffusion framework that denoises dance sequences at both section and interval levels. The architecture is shown in Fig. 2.

At the section-level, the model leverages section-level music features $\mathbf{M}_s$, structural labels $\mathbf{c}_s$, and denoised phase latents from adjacent sections $\overleftarrow{\mathbf{P}_s^0}, \overrightarrow{\mathbf{P}_s^0}$ to denoise the phase latent sequence $\mathbf{P}_s$. This enables the model to capture transitions in the musical structure and generate section-level guidance that is coherently aligned with these changes. The denoised $\mathbf{P}_s^0$ serves as high-level guidance for the subsequent interval-level diffusion process.

At the interval-level, we construct ControlNet modules to enable multi-aspect control in the periodic latent diffusion process. Specifically, interval-level music features $\mathbf{m}_i$, structural labels $\mathbf{c}_s$, denoised periodic latents from adjacent segments $\overleftarrow{\mathbf{q}_i^0}, \overrightarrow{\mathbf{q}_i^0}$, and the denoised phase latent $\mathbf{P}_i$ from the section-level process are integrated to generate the periodic latent signals $\mathbf{q}_i^0$. These signals are then decoded by the ACTOR-PAE decoder $\mathcal{P}_D$ to produce the dance sequence $\mathbf{x}_i$.

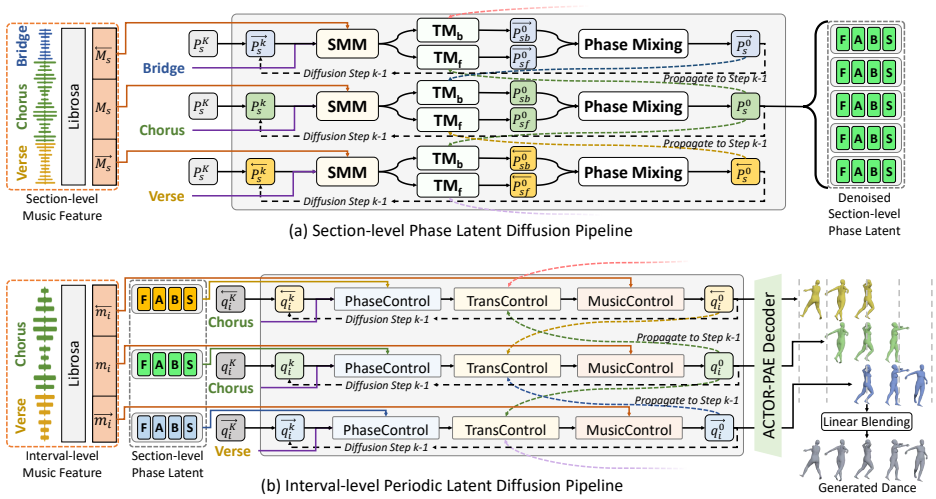


Figure 2: Overview of the two-level diffusion pipeline. (a) Section-level Phase Latent Diffusion: The phase latent sequence is denoised by the Section Music Module (SMM), conditioned on section-level music features. Transition Modules (TM_f and TM_b) incorporate context from adjacent sections. The resulting phase latent is passed to $\mathcal{D}_i^S$ for interval-level diffusion. (b) Interval-level Periodic Latent Diffusion: Periodic latent signals are denoised sequentially via PhaseControl ($\mathcal{D}_i^S$), TransControl ($\mathcal{D}_i^T$), and MusicControl ($\mathcal{D}_i^M$). The output is either further diffused or decoded by the ACTOR-PAE decoder to generate dance segments.

5.1 Section-level Phase Latent Diffusion

At the section-level, our goal is to denoise the phase latent sequence $\mathbf{P}_s$ to ensure coherence with the music structure and effectively respond to the music section transitions. The architecture is illustrated in Fig. 3(a). To achieve this, we first train a Phase Latent Diffusion Model $\mathcal{D}_s$ as the foundation for the phase latent diffusion process. The model is optimized using the loss function:

$$\mathcal{L}_{\mathcal{D}_s} = ||\mathbf{P}_s^0 - \mathcal{D}_s(k, \mathbf{c}_s, \mathbf{t}, \mathbf{P}_s^k)||_2, \tag{2}$$

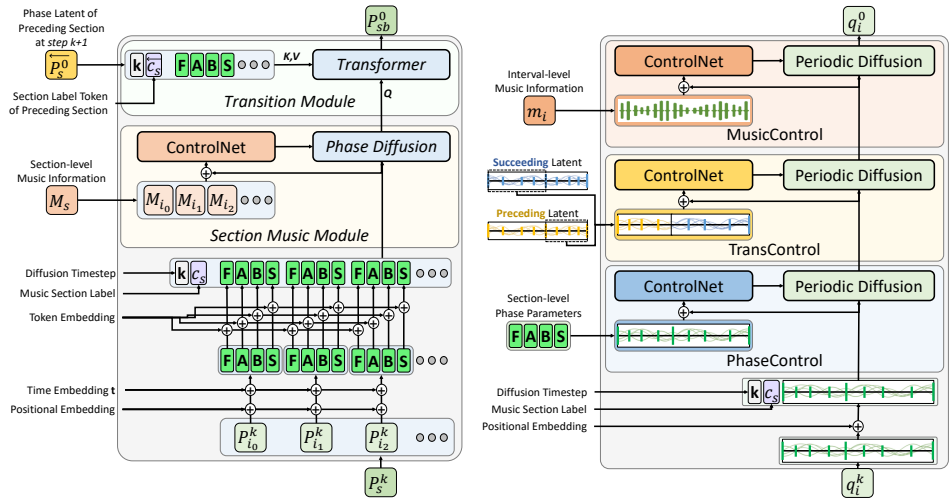
where k denotes the diffusion timestep, $\mathbf{c}_s$ is the music section label, and $\mathbf{t}$ indicates the temporal location of each phase latent relative to the center of the music section. The architecture of $\mathcal{D}_s$ is based on a self-attention Transformer Encoder, with input processing as illustrated in Fig. 3(a)(bottom). The positional and time embeddings $\mathbf{t}$ are shared among the phase parameter tokens within each $\mathbf{P}_i$, allowing the model to capture the duration of inter-beat intervals represented by each phase latent during the diffusion process. Additionally, four pretrained token embeddings are trained to distinguish between $[\mathbf{F}, \mathbf{A}, \mathbf{B}, \mathbf{S}]$ when they are concatenated temporally.

To enhance coherence with the music structure, we construct a Section Music Module (SMM) by incorporating section-level music features $\mathbf{M}_s$ into the diffusion process through a ControlNet module integrated with $\mathcal{D}_s$. Specifically, the music feature $\mathbf{M}_s = \{\mathbf{M}_i\}$ is processed to align with the phase parameter sequence $\mathbf{P}_s = \{\mathbf{P}_i\}$ during self-attention in the ControlNet module. The Section Music Module $\mathcal{D}_s^A$ is optimized using the loss function:

$$\mathcal{L}_{\mathcal{D}_s^M} = ||\mathbf{P}_s^0 - \mathcal{D}_s^M(k, \mathbf{c}_s, \mathbf{t}, \mathbf{P}_s^k, \mathbf{M}_s)||_2. \tag{3}$$

To enhance the model's responsiveness to music section transitions, we draw inspiration from TPDM Au et al. (2025) and integrate denoised phase latents from both the preceding section $\overleftarrow{\mathbf{P}_s^0}$ and the succeeding section $\overrightarrow{\mathbf{P}_s^0}$ into the diffusion process. For example, in the backward direction, we construct the Transition Module (TM_b), which performs cross-attention between the phase latent from the preceding section, $\overleftarrow{\mathbf{P}_s^0}$ (denoised at the previous diffusion step $k+1$), and the output phase latent from the Section Music Module at the current diffusion step, to generate the backward-conditioned phase latents $\mathbf{P}_{sb}^0$. The Transition Module $\mathcal{D}_s^B$ is optimized with the following loss:

$$\mathcal{L}_{\mathcal{D}_s^B} = ||\mathbf{P}_s^0 - \mathcal{D}_s^B(\mathcal{D}_s^M(k, \mathbf{c}_s, \mathbf{t}, \mathbf{P}_s^k, \mathbf{M}_s), \overleftarrow{\mathbf{c}_s}, \overleftarrow{\mathbf{t}}, \overleftarrow{\mathbf{P}_s^0})||_2. \tag{4}$$



(a) Details of Section-level Phase Diffusion Module (b) Details of Interval-level Periodic Diffusion Module

Figure 3: Details of the diffusion pipeline architecture. (a) Section-level Phase Diffusion Module: During the diffusion process of the phase latent sequence, the Section Music Module (SMM) integrates section-level music features, while the Transition Module (TM) incorporates information from adjacent sections. (b) Interval-level Periodic Diffusion Module: The denoising process of the phase latent sequence is sequentially conditioned on the section-level phase latent, periodic latents of adjacent segments, and music features using three ControlNets.

Note that both phase latents are preprocessed with their corresponding time embeddings $(t, \overleftarrow{t})$ and section labels $(\mathbf{c}_s, \overleftarrow{\mathbf{c}_s})$, as illustrated in Fig. 3(a)(bottom), before passing into cross-attention module.

After denoising both $\mathbf{P}^0_{sf}$ and $\mathbf{P}^0_{sb}$, we follow Au et al. (2025) and perform phase mixing to obtain the final phase latent $\mathbf{P}^0_s = \frac{\mathbf{P}^0_{sf} + \mathbf{P}^0_{sb}}{2}$. This phase latent is then either diffused back to step $k-1$ for further diffusion, or transferred to the interval-level diffusion pipeline at the final step.

5.2 Interval-level Periodic Latent Diffusion

At the interval-level, our objective is to denoise the periodic latent $\mathbf{q}_i$ under the guidance of music features, transition cues, and section-level phase latents. The corresponding architecture is depicted in Fig. 3(b). Similar to the section-level approach, we first train a Periodic Latent Diffusion Model $\mathcal{D}_i$ in the periodic latent space defined by the $\mathbf{q}_i$'s. The model is optimized using the following loss:

$$\mathcal{L}_{\mathcal{D}} = ||\mathbf{q}_i^0 - \mathcal{D}(k, \mathbf{c}_s, \mathbf{q}_i^k)||_2, \tag{5}$$

where the music section label $\mathbf{c}_s$ indicates the music section to which $\mathbf{q}_i$ belongs.

After training the diffusion model, we integrate ControlNet to incorporate additional control signals. Specifically, we propose three types of controls tailored for the interval-level diffusion process:

MusicControl: We incorporate the interval-level music segment $\mathbf{m}_i$ to denoise the associated periodic latent $\mathbf{q}_i$. The music-conditioned ControlNet $\mathcal{D}_i^M$ is optimized using the following loss:

$$\mathcal{L}_{\mathcal{D}_i^M} = ||\mathbf{q}_i^0 - \mathcal{D}_i^M(k, \mathbf{c}_s, \mathbf{q}_i^k, \mathbf{m}_i)||_2. \tag{6}$$

TransControl: We utilize the periodic latent from the preceding $\overleftarrow{\mathbf{q}_i^0}$ and succeeding segments $\overrightarrow{\mathbf{q}_i^0}$—both denoised at the previous diffusion step $k+1$—to denoise the current periodic latent $\mathbf{q}_i^k$. The interval-level segmentation follows the approach in Au et al. (2025), where the last 4 intervals of the preceding latent overlap with the first 4 intervals of the current latent. To form a transition-aware control signal $\mathbf{q}_{ps}$, we concatenate the last 4-interval of $\overleftarrow{\mathbf{q}_i^0}$ with the first 4-interval of $\overrightarrow{\mathbf{q}_i^0}$. This signal guides the current periodic latent $\mathbf{q}_i^k$ to maintain consistency with its overlapping context of nearby segments. The transition-conditioned ControlNet $\mathcal{D}_i^T$ is optimized using the following loss:

$$\mathcal{L}_{\mathcal{D}_i^T} = ||\mathbf{q}_i^0 - \mathcal{D}_i^T(k, \mathbf{c}_s, \mathbf{q}_i^k, \mathbf{q}_{ps})||_2. \tag{7}$$

PhaseControl: We leverage the denoised phase latent $\mathbf{P}_i \in \mathbf{P}_s$ obtained from section-level diffusion to assist in denoising the periodic latent $\mathbf{q}_i$ at the interval-level. This approach aims to facilitate smooth motion transitions between adjacent segments. During inference, we first reparameterize the predicted phase latents $\mathbf{P}_i$ to obtain the estimated periodic latent $\hat{\mathbf{q}}_i$, which is then provided as an additional input to the ControlNet $\mathcal{D}_i^S(k, \mathbf{c}_s, \mathbf{q}_i^k, \hat{\mathbf{q}}_i)$. During training, since $\mathbf{q}_i$ and the estimated latent $\hat{\mathbf{q}}_i$ are identical, we randomly mask $1 - \frac{k}{K}$ of the tokens in $\mathbf{q}_i$, excluding the nine beat onset frames at the boundaries of the 8 inter-beat intervals, to create $\hat{\mathbf{q}}_i$. This strategy prevents the network from trivially copying the input, where k and K are the current and total denoising steps, respectively. The section-level phase ControlNet $\mathcal{D}_i^S$ is optimized using the following loss:

$$\mathcal{L}_{\mathcal{D}_i^S} = ||\mathbf{q}_i^0 - \mathcal{D}_i^S(k, \mathbf{c}_s, \mathbf{q}_i^k, \hat{\mathbf{q}}_i)||_2. \tag{8}$$

After training the ControlNets, the denoising of interval-level motion segments $\mathbf{q}_i^k$ during inference is performed sequentially, as illustrated in Fig. 2(b). Specifically, for each diffusion step k, we first denoise the periodic latent $\mathbf{q}_i^k$ to $\mathbf{q}_i^0$ using $\mathcal{D}_i^S$, and then diffuse back to $\mathbf{q}_i^k$. Subsequently, $\mathbf{q}_i^k$ is processed by $\mathcal{D}_i^T$, followed by $\mathcal{D}_i^M$. The resulting $\mathbf{q}_i^0$ is either diffused back to step $k-1$, or decoded by the ACTOR-PAE decoder $\mathcal{P}_D$ to generate segment $\mathbf{x}_i$. Following Au et al. (2025), the generated segment is linearly blended with adjacent segments to produce a longer, coherent sequence.

5.3 Test-time Optimization for Foot-Skating Refinement

Foot contact artifacts can significantly undermine the realism of generated dance motion. To address this issue, we introduce a foot-skating loss function $\mathcal{FC}$, derived from the foot contact score in GMD Karunratanakul et al. (2023). This loss can be seamlessly integrated into the test-time iterative optimization process from SOSControl Au et al. (2026), enabling effective suppression of foot-skating artifacts in the generated sequences as a post-processing step after interval-level diffusion. Specifically, at each optimization step, we decode $\mathbf{q}_i^0$ into motion $\hat{\mathbf{x}_i^0}$ using ACTOR-PAE decoder $\mathcal{P}_D$, apply forward kinematics to obtain foot positions, and compute the foot horizontal speeds $\mathbf{s}_{xz}(\hat{\mathbf{x}_i^0})$ and foot vertical positions $\mathbf{p}_y(\hat{\mathbf{x}_i^0})$. These values are then used to calculate $\mathcal{FC}$ as follows:

$$\mathcal{FC}(\mathbf{q}_i^0) = \mathbf{s}_{xz}(\hat{\mathbf{x}_i^0}) * \sigma(\frac{\mathbf{p}_y(\hat{\mathbf{x}_i^0}) - \epsilon_p}{\tau}). \tag{9}$$

Here, ϵ_p denotes the vertical position threshold for foot contact estimation. The sigmoid function σ with temperature τ is used to generate a smooth foot contact signal, thereby enabling gradient backpropagation to optimize $\mathbf{q}_i^0$. Finally, we minimize $\mathcal{FC}$ using gradient descent:

$$\mathbf{q}_i^0 = \mathbf{q}_i^0 - \mathcal{M}_{fc}(\mathbf{q}_i^0). \tag{10}$$

As experimented in Au et al. (2026), iterative optimization through ACTOR-PAE decoder enables gradients from sparse guidance to propagate to adjacent frames, which preserves motion consistency.

6 Experiment

6.1 Dataset and Evaluation Metrics

We train and test our method, including comparison and ablation experiments, on the FineDance Li et al. (2023) dataset, which contains 7.7 hours of dance data in SMPL Loper et al. (2015) format at 30 FPS. The average dance length in FineDance is 152.3 seconds, providing substantial variation in music structure across long music tracks. This makes it an optimal benchmark for testing long dance sequence generation. Our experiments are conducted on the 20 music tracks in the FineDance test set. Note that the number of sections and 8-interval segments varies according to the length of each music track. The generated segments are then linearly blended to produce a continuous dance sequence that aligns with the full length of the input music. To further assess the robustness of HEDGE across different dataset distributions, we experimented on the JoruriPuppet dataset Dong et al. (2025), which contains performances of Japanese Ningyo Joruri puppets paired with music. Unlike existing human dance datasets Li et al. (2023; 2021), JoruriPuppet focuses on tempo-aware puppet movements, offering unique modeling challenges for music structure analysis.

Table 1: Evaluation on FineDance dataset. **Bold** value and underlined value indicate the best and the second-best performance, respectively.

Methods	$FID_k \downarrow$	$FID_g \downarrow$	$Div_k \rightarrow$	$Div_g \rightarrow$	BAS $\rightarrow$	FSR $\downarrow$	SJHK $\uparrow$
Real data	-	-	8.816	7.771	0.212	6.62	-0.008
FACT Li et al. (2021)	93.646	63.883	4.160	9.193	0.246	31.9	-0.057
Bailando Siyao et al. (2022)	91.229	47.403	7.020	9.668	0.224	19.7	0.025
EDGE Tseng et al. (2023)	89.099	48.789	5.104	5.700	0.265	35.8	-0.015
LODGE Li et al. (2024)	87.519	47.827	4.621	5.991	0.259	4.31	0.066
SoulNet Li et al. (2025)	163.76	160.01	2.497	4.058	0.220	11.8	0.014
HEDGE	**60.310**	**18.350**	**6.749**	6.973	**0.215**	7.53	0.115
HEDGE (FC opt.)	75.268	19.942	6.019	**7.132**	0.221	**4.16**	**0.116**

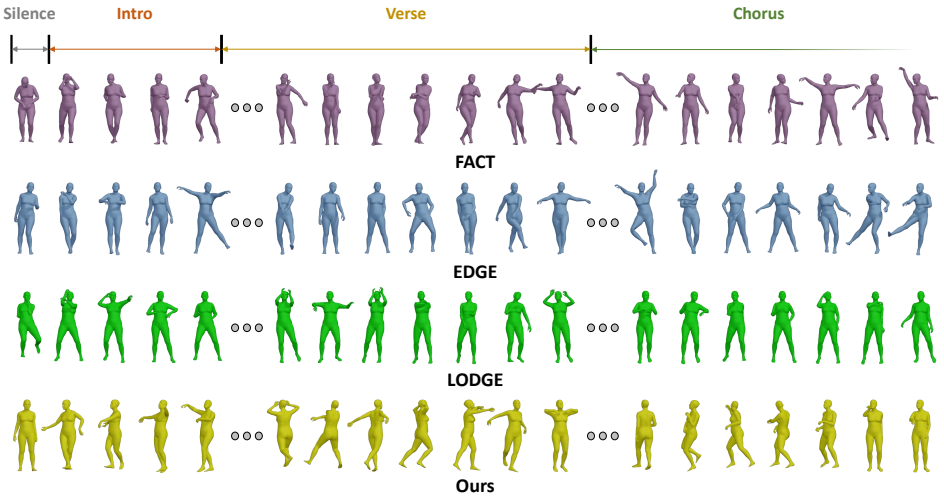


Figure 4: Qualitative comparison on the FineDance dataset. Top: section labels from the input music. Bottom: results are rendered in key frames. Notably, our method produces distinct motion contexts across different musical sections, whereas the baseline methods tend to generate uniform motion content across various sections.

For the evaluation metrics for music tempo alignment, we propose the Section-aware Jo-Ha-Kyu (SJHK) metric, inspired by the Jo-Ha-Kyu (JHK) metric from JoruriPuppet Dong et al. (2025). The original JHK assesses alignment by correlating overall average joint-speed with instantaneous musical tempo, yielding a 1-dimensional speed sequence and a 1-dimensional tempo sequence. The Pearson correlation between these two sequences is then evaluated to obtain the JHK score. However, a single correlation over long tracks with multiple sections tends to smooth out section-level variations and is dominated by global linear trends of uniform tempo. To address this, SJHK segments the speed and tempo sequences using music structure, computes the Pearson correlation within each section, and averages the results. This section-aware design better captures alignment to structural tempo changes over long pieces. For rhythmic alignment, we use the Beat Alignment Score (BAS), defined as the mean distance between each music beat and its nearest motion beat. We rate BAS against the reference performance so that scores are capped at the reference level, discouraging unnaturally dense or hyperactive motions that might appear to outperform the reference testing motion.

Beyond alignment, we also evaluate other metrics, including FID_k, FID_g, DIV_k, DIV_g, and FC, following the protocol established in LODGE Li et al. (2024). Specifically, the Fréchet Inception Distance (FID) is employed to evaluate the motion quality by computing the distribution difference between motion features extracted from generated outputs and ground-truth sequences. Two variants of FID are used: FID_k, computed on kinematic features, measures the physical realism of the motion, while FID_g, based on geometric features, assesses the overall geometric quality. In addition, we report DIV_k and DIV_g, which quantify the diversity within the kinematic and geometric feature distributions. The Foot Skating Ratio (FSR) is utilized to assess the presence of foot-skating artifacts, measuring the proportion of frames in which either foot slides while touching the ground.

6.2 Long Dance Generation Result

We compare the performance of our method with existing music-to-dance generation models, including FACT Li et al. (2021), Bailando Siyao et al. (2022), EDGE Tseng et al. (2023), LODGE Li et al. (2024), and SoulNet Li et al. (2025). FACT, Bailando, and EDGE generate long dance sequences through autoregressive modeling. In contrast, LODGE first generates global-level guidance over a window of 1024 frames, followed by diffusion in a smaller window to refine rhythmic details. We assess our proposed HEDGE framework with and without foot contact iterative optimization as a post-processing step. As shown in Tab. 1, HEDGE outperforms competing methods on the FID, DIV, and SJHK metrics, and attains notably improved FSR with foot contact post-processing. HEDGE achieves the closest BAS to the reference motion. This highlights HEDGE's ability to generate diverse and realistic dance motion, aligning effectively with changes in music structure throughout long music tracks. In contrast, FACT and EDGE exceed the reference on BAS, indicating an overemphasis on beat synchronization at the expense of motion realism and structural coherence. LODGE attains moderate SJHK due to its global guidance and achieves reasonable FSR via the Foot Refine Block, which incorporates forward kinematics during diffusion, but its lack of explicit musical-structure modeling limits performance. SoulNet's BAS and SJHK are close to the reference, but FID scores are high due to incompatibility between the HRVQ module and LODGE's body-pose evaluation protocol.

Fig. 4 provides a qualitative comparison on the FineDance dataset. Section labels from the input music are shown at the top, while the generated dance sequences are visualized as key frames at the bottom. Our method consistently demonstrates fine-grained sensitivity to musical structure. Specifically, HEDGE remains stationary during silent sections, faithfully reflecting musical pauses, whereas baseline methods such as LODGE and EDGE continue to generate motion. Additionally, our approach produces contextually appropriate and diverse movements across different musical sections, such as smooth, pleasing motions during the intro and more rapid, dynamic movements in the chorus. In contrast, baseline models tend to produce uniformly rapid movements throughout, optimizing only for beat alignment and neglecting section-specific musical cues. These findings visually confirm the effectiveness of our hierarchical framework and section-aware evaluation metrics, which together enable HEDGE to generate dance sequences that are not only beat-synchronized but also structurally aligned with the underlying musical progression.

6.3 Additional Evaluation and User Studies

Beyond the main experiments on HEDGE, we conducted additional ablation studies to evaluate the contribution of individual modules and to analyze model performance under varying hyperparameter settings. We also performed a user study, where Participants assessed each rendered dance sequence based on three criteria: Naturalness of Movements, Motion-Music Rhythmic Coherence, and Motion-Music Structure Alignment. Results indicate that HEDGE is consistently preferred across all evaluation criteria. Additional details regarding the user study and comprehensive ablation results are provided in the supplementary material.

6.4 Conclusion

In this work, we tackle the challenge of generating dance that reflects the full temporal and structural complexity of music. We propose HEDGE, a two-level diffusion model developed upon a hierarchical organization of both music and motion. Our framework segments both music and motion at the section and interval levels, and encodes motion using phase and periodic latents. This hierarchical approach enables long-term alignment with musical structure while maintaining detailed, musically synchronized choreography at a fine-grained level. Through comprehensive experiments utilizing section-aware evaluation metrics, we demonstrate that HEDGE not only produces realistic long-term dance movements but also effectively captures expressive dynamics across different musical sections. Our approach narrows the expressive gap between music structure and dance choreography, laying a solid foundation for further research in automated choreography, virtual character animation, and broader music-driven motion synthesis applications.

AI USE STATEMENT

In this work, we have not used generative AI tools for design or evaluate the research methodology or experiments; implement methods; translate; clean or reformat data; conduct qualitative or thematic analyses; or interpret results, and generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, contributing to proofs, or proposing/refining hypotheses are not applicable to this work. Additionally, we used generative AI tools to summarize and analyze prior literature, and edit the manuscript for clarity and readability. We have reviewed all AI-assisted work. We manually verified and revised LLM-generated paper writing content for coherence and accuracy, including the descriptions of related work and methodology. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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

Ho Yin Au, Jie Chen, Junkun Jiang, and Yike Guo. Choreograph: Music-conditioned automatic dance choreography over a style and tempo consistent dynamic graph. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 3917–3925, 2022.

Ho Yin Au, Jie Chen, Junkun Jiang, and Jingyu Xiang. Deep compositional phase diffusion for long motion sequence generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Ho Yin Au, Junkun Jiang, and Jie Chen. Soscontrol: Enhancing human motion generation through saliency-aware symbolic orientation and timing control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(24):19632–19639, Mar. 2026. doi: 10.1609/aaai.v40i24.39043. URL https://ojs.aaai.org/index.php/AAAI/article/view/39043.

Morgan Buisson, Brian Mcfee, and Slim Essid. Using pairwise link prediction and graph attention networks for music structure analysis. In *International Society for Music Information Retrieval (ISMIR)*, 2024.

Ran Dong, Shaowen Ni, and Xi Yang. Joruripuppet: Learning tempo-changing mechanisms beyond the beat for music-to-motion generation with expressive metrics. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–11, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Korrawe Karunratanakul, Konpat Preechakul, Supasorn Suwajanakorn, and Siyu Tang. Guided motion diffusion for controllable human motion synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2151–2162, 2023.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

Lucas Kovar, Michael Gleicher, and Frédéric Pighin. Motion graphs. *ACM Transactions on Graphics*, 21(3):473–482, 2008.

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*, pp. 10234–10243, 2023.

Ronghui Li, YuXiang Zhang, Yachao Zhang, Hongwen Zhang, Jie Guo, Yan Zhang, Yebin Liu, and Xiu Li. Lodge: A coarse to fine diffusion network for long dance generation guided by the characteristic dance primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1524–1534, 2024.

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.

Xiaojie Li, Ronghui Li, Shukai Fang, Shuzhao Xie, Xiaoyang Guo, Jiaqing Zhou, Junkun Peng, and Zhi Wang. Music-aligned holistic 3d dance generation via hierarchical motion modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14420–14430, 2025.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P 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*, volume 8, pp. 18–25, 2015.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 1530–1538, 2015.

Yoni Shafir, Guy Tevet, Roy Kapon, and Amit Haim Bermano. Human motion diffusion as a generative prior. In *The Twelfth International Conference on Learning Representations*, 2024.

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.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Sebastian Starke, Ian Mason, and Taku Komura. Deepphase: Periodic autoencoders for learning motion phase manifolds. *ACM Transactions on Graphics*, 41(4):1–13, 2022.

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SJ1kSyO2jwu.

Jonathan Tseng, Rodrigo Castellon, and 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.

Guillermo Valle-Perez, Gustav Eje Henter, Jonas Beskow, Andre Holzapfel, Pierre-Yves Oudeyer, and Simon Alexanderson. Transflower: Probabilistic autoregressive dance generation with multimodal attention. *ACM Transactions on Graphics*, 40(6):1–14, 2021.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017.

Zhao Yang, Bing Su, and Ji-Rong Wen. Synthesizing long-term human motions with diffusion models via coherent sampling. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 3954–3964, 2023.