

Transformer-Based Two-level Approach for Music-driven Dance Choreography

Yanbo Cheng^{1a} and Yingying Wang^{1b}

Department of Computing and Software, McMaster University, Hamilton, Canada

Keywords: Deep Learning, Character Animation, Motion Synthesis, Motion Stylization, Multimodal Synchronization.

Abstract: Human dance motions are complex, creative, and artistic expressions. Synthesizing high-quality dance motions and synchronizing them to music has always been a challenge in animation research. Three problems in synthesizing dance motions are presented: 1) dance movements are complex non-linear motions that follow high-level structures of the dance genre over a long horizon, yet must maintain a stylistic consistency; 2) even for the same genre, dance movements require diversity, expressiveness, and nuances to appear natural and realistic; 3) spatial-temporal features of dance movements can be influenced by music. In this paper, we address these issues using a novel two-level transformer-based dance generation system that can synthesize dance motions to match the audio input. Our high-level transformer network performs the choreography and generates dance movements with consistent long-term structure, and our low-level implementer infuses diversity and realizes actual dance performances. This two-level approach not only allows us to generate dances that are consistent in structure, but also enables us to effectively add styles learnt from a wide range of dance datasets. When training the choreography model, our approach fully utilizes existing dance datasets, even those without musical accompaniment, and thus differs from previous research that requires dance training data to be accompanied by music. Results in this work demonstrate that our two-level system generates high-quality dance motions that flexibly adapt to varying musical conditions trained on a dataset of dance sequences without accompanying music.


1 INTRODUCTION


Dance movements are complex artistic human expressions. Automatic generation of dance motions that synchronize to musical beats and match a musical style, can assist an artist's manual choreography, and can provide automatically synthesized dance for virtual concert, augmented reality, and other forms of digital entertainment. Further, by combining audio integration in platforms like the metaverse, artistic human composition and expression can be fully realized and can benefit from automatic dance choreography.

In this work, we choreograph dance motions by using a novel two-level transformer-based approach, where the high-level choreographer learns to model the long-term movement structure of the dance genre and the low-level implementer incorporates the music influence and realizes the actual dance motions with fine details and nuances. The advantages of using the two-level system are that 1) the dance mo-

tions are decoupled from the music and thus have less restrictive training data requirements which make learning a model possible by fully utilizing standalone dance datasets without musical accompaniment; 2) the dance structure learning is separate from the motion implementation, which allows variations in different granularity levels, i.e. output dance can either be structurally different or stylistically different given input music; 3) the choreograph is greatly accelerated as the automated choreographer focuses on capturing high-level core structure of the dance genre; 4) and lastly, the music rhythm and energy can be customized in the low-level implementation, which allows for better user control.

Synthesizing high-quality dance motion is a great challenge in animation research. Unlike locomotion which has cyclic movement with well-defined phase patterns, dance motions have more complex time signatures with expressive, creative, and artistic compositions. On the one hand, dance movements of any genre follow their high-level dance structure. On the other hand, performances from different dancers, or

^a  <https://orcid.org/0009-0009-1684-0585>

^b  <https://orcid.org/0000-0002-5680-1929>

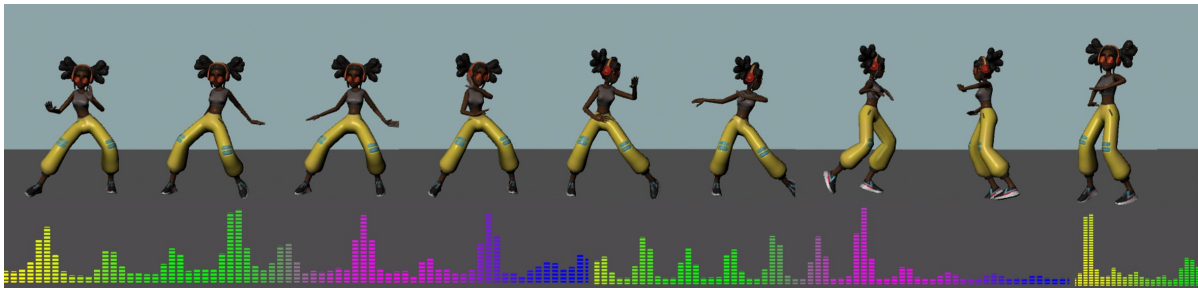


Figure 1: Two-level dance choreography approach decouples dance structure learning from its diverse implementation. The high-level choreographer predicts the next dance segment, and the low-level implementer fills in expressive variations given audio input.

from the same dancer but in different takes, or influenced by different music can have stylistic variations. Data-driven motion synthesis becomes even harder, if not entirely intractable, due to the vast combination of motion content and motion styles (Smith et al., 2019; Xia et al., 2015; Yumer and Mitra, 2016). Furthermore, the influence of audio channels makes dance synthesis a multi-modal problem. Advanced deep-learning models have been widely used in predicting language sequences. However, training sequence prediction models for dance choreography has been restricted because of scarce 3D dance data, most of which is unpaired with music.

We regard dance as a highly complex form composed of unique body language signatures. Dance movements follow a specific high-level motion syntax, have independent expressions, and can be influenced by music during a performance, but not a byproduct of music. Our research fully utilizes existing dance datasets (Morro Motion, 2017), to learn a dance syntax without the need for paired accompanying music. During data processing, long sequences of dance motions are split into segments, where each segment represents a dance word that contains either a motion stroke emphasis or connection for a dance composition. Dance segments are further passed through pre-trained autoencoder networks to extract their embeddings and cluster into synonym groups based on the similarity of embeddings. To predict the long-horizontal structure of the dance genre, a transformer-based model is trained as the high-level choreographer. The choreographer takes the previous dance segments’ embeddings as input and predicts the next dance segment embedding. The predicted embedding is further passed to the low-level dance implementer to incorporate the influence of music and infuse variations into the dance performance. The low-level dance implementer consists of a feature matcher which is responsible for finding a dance segment that matches the music style; and a dance synchronizer which is responsible for synchro-

nizing the dance segment to the music beat. The feature matcher takes the dance segment embedding predicted by the high-level choreographer, identifies the nearest synonym cluster, and within the cluster selects the dance segment that best matches the music rhythm and energy features. The synchronizer takes the selected dance segment and the detected musical beats as input, adjusts the dance segment timing to align its motion emphasis with the music beat, and blends it into the final performance.

We summarize the main contribution of our work as follows:

- We demonstrate a novel transformer-based sequence model to solve dance choreography as a body language composition problem;
- We present a two-level approach to decouple the high-level dance structure from its low-level realization to ensure structure consistency and realization diversity;
- We propose a strategy that separates motion learning from multi-modal synchronization, and thus can take full advantage of existing dance datasets, even unpaired with accompanying music.

2 RELATED WORK

Deep Learning for Human Motion. With the rapid development of deep learning, researchers have successfully applied deep neural network architectures to human motion learning, control, and synthesis. Early work (Taylor and Hinton, 2009) uses a factored Conditional Restricted Boltzmann Machine (CRBM) to model human motions. To better understand human motions, deep autoencoders (Holden et al., 2015; Wang and Neff, 2015) have been used to extract high-level motion features. By stacking task-specific networks on top of autoencoder networks, Holden et al. (Holden et al., 2016) proposed a deep learning framework that is capable of performing motion edit-

ing, control, and stylization. Based on fully connected feed-forward network structures with gating control, Phase-Functioned Neural Networks (PFNN) (Holden et al., 2017) successfully synthesized human locomotion in real-time adapting to the different terrains and following users' instructions. The follow-up work Mode-Adaptive Neural Networks (Zhang et al., 2018a) can even handle more complicated foot patterns and synthesize varied quadruped locomotion.

Regarding human motion as temporal sequences, researchers have explored Recurrent Neural Network (RNN) structures to model human motions. Li et al. (Li et al., 2017) proposed a novel auto-conditioned Long Short-Term Memory (LSTM) structure. To predict long-horizon human motions, Wang et al. (Wang et al., 2019) used Spatio-Temporal Recurrent Neural Networks (STRNN) to model spatial and temporal variances in human motions and disambiguate the pose prediction. Deep reinforcement learning (DRL) has been applied to motion skill learning, including locomotion (Peng et al., 2017), balancing skills (Liu and Hodgins, 2017), basketball dribbling (Liu and Hodgins, 2018), and acrobat stunts (Peng et al., 2018). Researchers have also explored DRL to simulate physical styles, by considering muscle strength, body proportions, and environmental conditions (Lee et al., 2021). Based on the generative adversarial network (GAN), Peng et al. (Peng et al., 2021) integrated Adversarial Motion Priors (AMP) into DRL as the discriminator for generating a wider range of physics-based human motions. Aberman et al. (Aberman et al., 2020) proposed a generative model that is capable of extracting motion styles from videos and generating stylized human motions accordingly.

Audio-Driven Motion Synthesis. Researchers have explored approaches to utilize audio input to drive, disambiguate, and stylize motion generation. By using rule-based methods or statistical methods like Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs), early research extracts prosody or verbal features to predict body language (Levine et al., 2009), gestures (Levine et al., 2010), head orientation (Sargin et al., 2007), lip movement (Englebienne et al., 2007; Park and Ko, 2008) and facial expressions (Albrecht et al., 2002; Chuang and Bregler, 2005). In recent years, deep learning methods have been successfully applied to predict multi-modal behaviors. Ferstl et al. (Ferstl et al., 2019) used LSTM-based networks as a phase classifier and Gated Recurrent Unit (GRU)-based networks as a gesture generator. Yoon et al. (Yoon et al., 2020) proposed a multi-layered bidirectional GRU network with an adversarial discriminator to predict

gestures from audio, text, and speaker identity. A bidirectional LSTM model is used in (Hasegawa et al., 2018) to learn the speech-gesture relationships for gesture synthesis. Habibie et al. (Habibie et al., 2022) proposed a motion matching-based framework that selects clips using a Nearest Neighbor-based algorithm and further synthesizes gestures through a conditional GAN in a controllable way. In addition to conversational human motions, music spectral features have been used to drive arm and finger animations playing musical instruments. Given a piece of music or a midi file, optimization-based procedural methods are developed to generate finger movements performing piano (Zhu et al., 2013) and guitar (ElKoura and Singh, 2003). An LSTM-model (Shlizerman et al., 2018) is trained on piano and violin videos, which takes audio input, and predicts the performance animations for virtual avatars.

Dance Motion Choreography. Dance motions, different from playing musical instruments or performing multi-modal gestures, have much more complicated movement flow and patterns and thus have been a great challenge in computer animation research. Early research (Okamoto et al., 2010) temporally scales leg motions during dancing to match the tempo of music. Shiratori et al. (Shiratori et al., 2006) synthesizes dance motions by selecting dance segments that match the music rhythms and intensity. Aristidou et al. (Aristidou et al., 2017) uses Gaussian Radial Basis Function (RBF) to model the correlation between dance motion features and emotion coordinates. Aristidou et al. (Aristidou et al., 2021) samples the probability distribution of dance motifs to maintain a consistent global dance structure, and further infuses styles using AdaIn layers and implements motions through an acLSTM network. Other researchers put more emphasis on choreography controllability by proposing a dance show authoring platform (Schulz and Velho, 2011), developing a body-part motion synthesis system (Soga et al., 2016) or Disk Jockeys (DJ) user interface (Iwamoto et al., 2017) to incorporate human input during choreograph.

In recent years, many deep learning models have been proposed to solve dance choreography problems. Alemi et al. (Alemi et al., 2017) trained Factored Conditional Restricted Boltzmann Machines (FCRBM) to predict dance motions frame by frame while synchronizing to musical input. Li et al. (Li et al., 2020) developed a two-stream transformer generative model which takes the motion and audio as input and fuses their features to synthesize diverse dance motions. Li et al. (Li et al., 2021) uses three

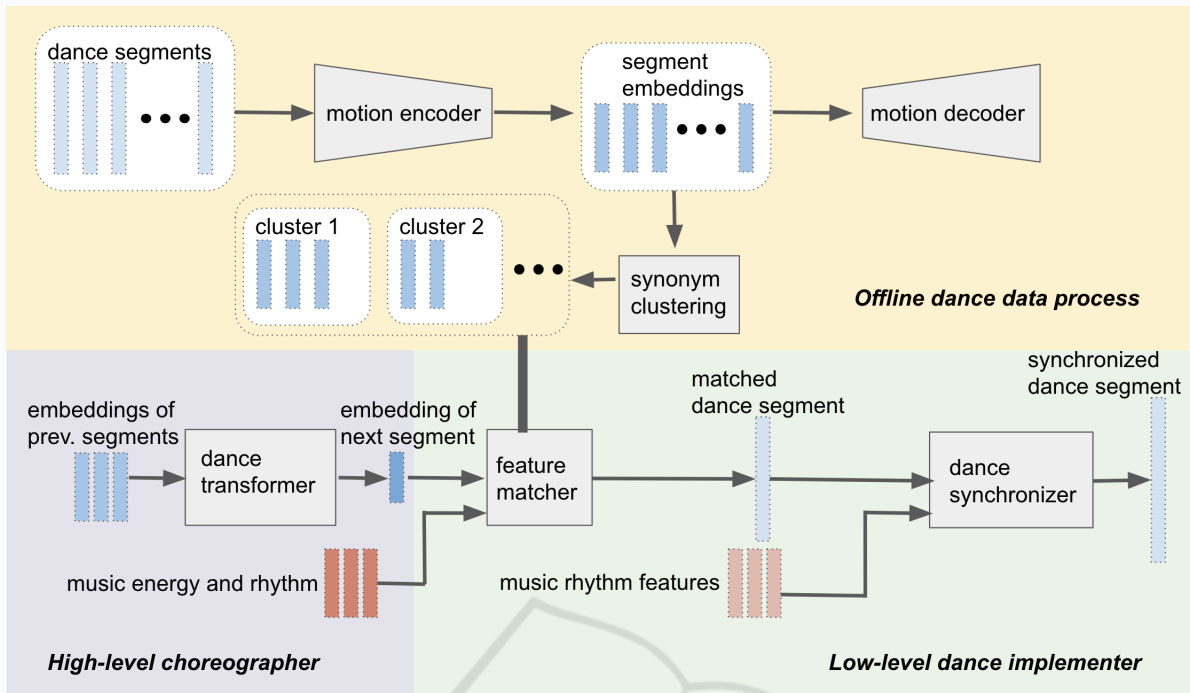


Figure 2: System Overview.

transformers for extracting motion, music, and cross-modal features, and auto-regressively generates long-horizon dance motions. To model the relationship between music and dance, Chen et al. (Chen et al., 2021) extracts choreomusical style and rhythm embeddings, maps them together with motion embeddings to a unified space, and generates the dance motions through graph-based optimization. Lee et al. (Lee et al., 2019) models dance variations using a Dance Unit Variational Auto Encoder (DU-VAE) and generates a sequence of dance movements given music input using a Music-to-Movement GAN during composition. Siyao et al. (Siyao et al., 2022) trained an actor-critic Generative Pre-trained Transformer (GPT) model using dance poses in a large dance motion dataset (Li et al., 2021). Alexanderson et al. (Alexanderson et al., 2023) trained a model using diffusion methods, which adapts the DiffWave architecture to represent three-dimensional pose sequences, incorporating Conformers for enhanced performance.

3 OVERVIEW

Our dance synthesis workflow is illustrated in Figure 2. Given a dance dataset, we preprocess the motions offline. Long sequences of dance motions are first cut into shorter dance segments where each segment

maximally contains one motion emphasis. Temporal phase information is then labelled in a semi-automatic way for the dance segments. We regard dance segments as the basic words of body language. Segments are then passed to an autoencoder network pre-trained on a large 3D human motion dataset (Holden et al., 2016) to extract the corresponding dance word embeddings. Based on the similarity of word embeddings, dance segments are grouped into synonym clusters, where within each cluster, segments are categorized as the same content but with performance variations. While between different clusters, dance segments have different semantic meanings.

The high-level choreographer is a transformer-based sequence model trained on the preprocessed dance dataset. It takes the previous dance segments' embeddings as input and predicts the embedding of the next segment. Though dance implementations and styles can be influenced by music and individual performers, dance itself is a form of expression in the body language channel. Thus, the high-level choreographer is trained only on motion data, to focus on the learning of syntactic structure of a dance genre without over-fitting to variations in the actual implementation.

The low-level dance implementer, which consists of a motion feature matcher and a dance synchronizer, is responsible for infusing variations into the performance. The feature matcher takes the predicted em-

bedding from the high-level choreographer and finds its nearest synonym cluster to have a rich pool of dance segments for motion implementation. Within the synonym cluster, the feature matcher selects the dance segment that best matches the input music energy and rhythm. The feature matcher is decoupled from the high-level dance structure prediction and helps generate diverse dance motions under musical influence, yet still respects the global structure of the dance genre. The dance synchronizer further synchronizes the motion emphasis in the selected dance segment to the music beat, based on the segment’s temporal phase information and the music’s rhythmic features.

4 METHOD

4.1 Data Processing

Our approach requires three-dimensional dance motion datasets to train the two-level choreographer model. Compared to previous work, training of the two-level choreographer works with silent motions, i.e. dance motions without accompanying music. This relaxation in data requirements qualifies a great amount of motion capture resources available as training-sets, and thus significantly reduces the workload for motion collection. While many public dance motion capture databases suffice the requirement, for this study, we acquired a dance dataset (Morro Motion, 2017), consisting of 61 long sequences and totalling 37,983 frames of dance motions (with no music). Basic data augmentation was performed by mirroring the motions, which doubles the data size.

4.1.1 Motion Segmentation and Phase Labelling

In the original dataset, motions are in long continuous performances with connected dance moves. We cut the long motion sequences into smaller dance segments, where each segment is a base unit for composing longer sequences. Regarding dance as a form of body language, we adopted a similar concept of gesture phrase segmentation based on previous gesture synthesis work (Neff et al., 2008; Levine et al., 2009; Smith and Neff, 2017; Levine et al., 2010; McNeill, 2008). Specifically, we followed the motion segmentation routine such that each segment has at most one motion emphasis, e.g. arm stroke or footstep. The temporal phase of each segment follows the general pattern $[preparation, emphasis, retraction]$, and is denoted by $p \in [0, 1]$. A time function $T(p)$ is defined as below to map the input phase p to the index

of corresponding frame in the dance segment:

$$T(p) = \begin{cases} 2p * t_e, & \text{if } p \in [0, \frac{1}{2}] \\ t_e + 2(p - \frac{1}{2}) * (t - t_e) & \text{if } p \in (\frac{1}{2}, 1] \end{cases} \quad (1)$$

When $p = 0$, it indicates the beginning of each dance segment, and p keeps increasing with the progress of motion until $p = \frac{1}{2}$ i.e. it reaches the motion emphasis t_e ; when $p \in (\frac{1}{2}, 1]$, it corresponds to motion retraction to the end of the segment. Dance segments that do not have an emphasis are regarded as connection motions, and their phase increases from 0 to 1, and maps to a frame index linearly. Motion segmentation and emphasis annotation were performed manually, and p values were computed automatically based on segment boundary and emphasis. The street dance dataset produced 1522 dance segments in total.

4.1.2 Embedding Extraction

Instead of directly using dance segments for choreographer training, we regard each segment as a word in a body language and thus perform a similar word embedding extraction process to that found in the Natural Language Process (NLP) domain.

Each motion segment is represented by $X = [\mathbf{x}_1, \dots, \mathbf{x}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_t]$, where \mathbf{x}_i the full body pose at frame i , and t indicates the number of frames of the motion segment. For each frame, $\mathbf{x} = [\mathbf{p}, \dot{\mathbf{p}}_{\text{root}}, \dot{\mathbf{r}}_{\text{root}}, \mathbf{c}] \in \mathbb{R}^d$ describes the full body pose, and thus X is in $\mathbb{R}^{t \times d}$ space. Specifically, $\mathbf{p} \in \mathbb{R}^{63}$ are the three-dimensional positions for 21 essential joints, $\dot{\mathbf{p}}_{\text{root}} \in \mathbb{R}^2$ are the root joint’s linear velocity in x and z directions, $\dot{\mathbf{r}}_{\text{root}} \in \mathbb{R}^3$ are the root joint’s angular velocity around xyz axes, $\mathbf{c} \in \mathbb{R}^2$ are left and right foot contact information respectively, and thus the dimension of full body pose is $d = 70$.

Segment embeddings are extracted using the autoencoder architecture proposed in the previous work by (Holden et al., 2016), and pre-trained on a large combined human motion dataset, including CMU Mocap (CMU, 2000), HDM05 (Müller et al., 2007), MHAD (Ofli et al., 2013), and Xia-Style (Xia et al., 2015). The forward pass of the autoencoder, i.e. the encoder, takes a segment X of t -frame long as input ($t = 240$ is used in the pre-training), performs the operation Φ , and projects X to high-level embeddings H in the hidden motion manifold:

$$H = \Phi(X) = \text{ReLU}(\text{MaxPool}(X * W_0 + b_0)) \quad (2)$$

With temporal max pooling, $H \in \mathbb{R}^{\frac{1}{2} \times h}$ includes $\frac{1}{2}$ frames of projected h -dimensional hidden units. W_0 is a $d \times h \times w$ weight matrix that converts each frame from d -dimensional motion space to h -dimensional

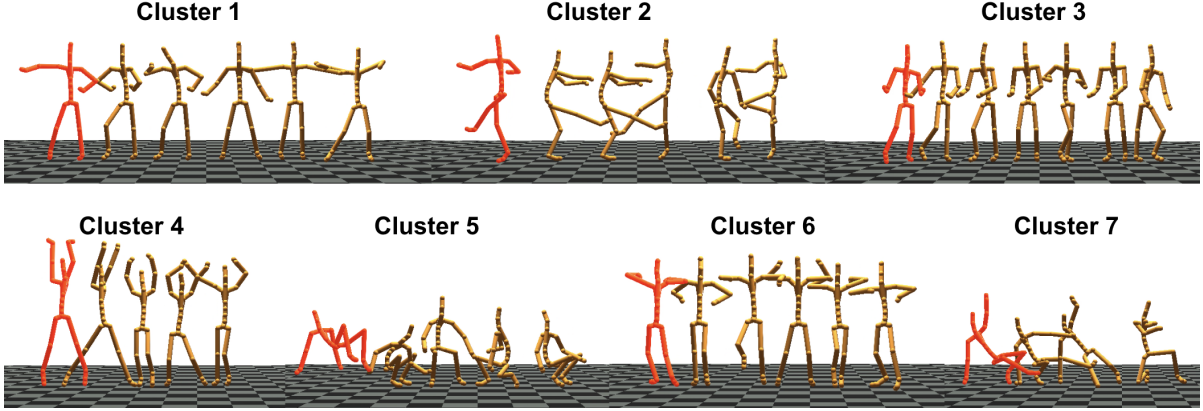


Figure 3: Dance segments clustered based on their extracted embeddings.

hidden space using temporal convolutional filters of size $w = 25$. b_0 is the h -dimensional bias vector. $h = 256$ is used in the pre-trained network.

The backward pass of the autoencoder, i.e. the decoder, performs operation Φ^+ , and maps the embedding H back to \hat{X} in observation motion space.

$$\hat{X} = \Phi^+(H) = (\text{InversePool}(H) - b_0) * \tilde{W}_0 \quad (3)$$

The autoencoder was trained with both a forward and a backward pass, to minimize the reconstruction error between X and \hat{X} . In this work, we use the forward encoder pass to extract motion segment embeddings. To fully utilize the pretrained autoencoder, we renormalized each dance segment to 240-frame length so that $X \in \mathbb{R}^{240 \times 70}$ fits in one pass and generates one consolidated embedding $H \in \mathbb{R}^{120 \times 256}$ for each segment. The extracted embeddings capture the spatial features of dance movements, and we leave the temporal synchronization to the synchronizer.

4.1.3 Synonym Clustering

For all the dance segments, we employed Gaussian Mixture Models (GMMs) to cluster similar segments based on their embeddings H on the hidden manifold and applied the Expectation-Maximization (EM) algorithm to estimate the model parameters so that the likelihood of the data is maximized. 1522 dance segments, represented by their embeddings, produced 21 clusters of various sizes, with large clusters typically containing around 100 segments, and small clusters containing about 10 ~ 20 segments. Though dance segments were clustered based on their embeddings, visually each cluster covers a set of similar dance movements with unique characteristics and diverse implementation (illustrated in Figure 3). We compute the mean embeddings of all segments included in one cluster as the cluster embedding \bar{H}_c , to represent each cluster.

4.2 High-Level Dance Choreographer

4.2.1 Model Architecture

High-level choreographer learns the long-horizon structure of dance movement and is implemented using a transformer-based network, originally proposed by (Vaswani et al., 2017) and (Wu et al., 2020). The network includes three encoder layers and decoder layers respectively (illustrated in Figure 4).

The encoder is made up of a positional encoding layer and three identical encoder layers. Compared with the original transformer structure in (Vaswani et al., 2017), we removed the input layer and directly plugged in the extracted h -dimensional hidden features. The positional encoding layer employs the two operations below

$$PE(i, 2j) = \sin\left(\frac{i}{10000^{2j/h}}\right) \quad (4)$$

$$PE(i, 2j+1) = \cos\left(\frac{i}{10000^{2j/h}}\right) \quad (5)$$

to inform the network about the relative position of the segment in the input sequence, where i, j are the position index and dimension index in the hidden space respectively. The output of the positional encoding layer is fed into the encoder, which includes a self-attention sub-layer and a fully connected feed-forward sub-layer.

The decoder is made up of three identical decoder layers and a linear mapping layer. Compared with the three encoder layers, in the three decoder layers, there are similar encoder-decoder attention sub-layers performing the multi-head attention to the output of the encoder stack. It allows every position in the decoder to attend to all positions in the input sequence. Outputs from the top decoder layer are passed to the linear mapping layer and converted to the target dance segment embedding.

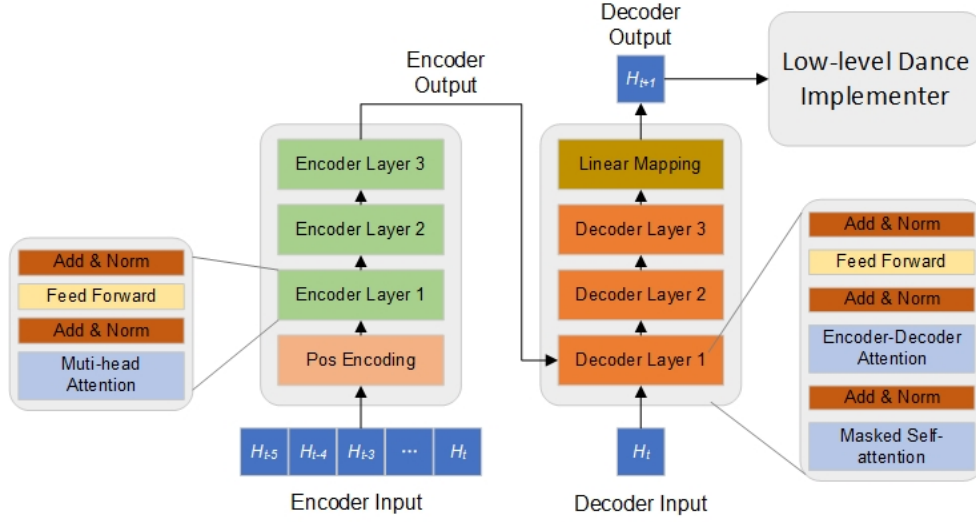


Figure 4: High-level choreographer implemented in a transformer-based network architecture.

4.2.2 Training

The input to the encoder is $[H_{t-5}, H_{t-4}, H_{t-3}, H_{t-2}, H_{t-1}, H_t]$, i.e. the embeddings of the current dance segment H_t , and its five previous dance segments. The input to the decoder begins with H_t and it learns to output H_{t+1} through supervised training. The dance segments' original order in the motion captured dataset is used as the ground truth. The loss function L is defined as the Mean Squared Error (MSE) between the ground truth dance segment embedding H_{t+1} and segment embedding \hat{H}_{t+1} predicted by the decoder:

$$L = \frac{1}{h} \sum_{d=1}^h \left(H_{t+1}^d - \hat{H}_{t+1}^d \right)^2 \quad (6)$$

During training, Adam optimizer (Kingma and Ba, 2014) with parameters $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$ and batch size 32 is used. The learning rate is initially set as 0.003, and scheduled to decay by the coefficient $e^{-0.05}$ each epoch. We trained the model for a total of 50 epochs.

4.3 Low-Level Dance Implementer

The goal of the low-level dance implementer is to realize various dance performances that match the accompanying music, given the dance segment embedding \hat{H}_{t+1} output from the high-level choreographer. It takes music and dance segment embedding \hat{H}_{t+1} as input, finds the best dance segment, and incorporates it into the long dance sequence.

4.3.1 Music Features

To generate a diverse dance performance that matches the music, we first extract two key features from music, i.e. beat and energy.

Music Beat Detection. We use Librosa (McFee et al., 2015) for detecting beats from music input. Librosa first computes the onset strength envelope (OSE) and estimates the music tempo. Music beats are tracked using a dynamic programming approach (Ellis, 2007) that seeks to maximize the alignment between the estimated tempo, the observed onset strengths, and the temporal consistency of inter-beat intervals. The algorithm iteratively refines beat timing, ensuring that the detected beats correspond to significant onsets and adhere to a consistent tempo.

To speed up beat detection in an online fashion for real-time application, we implemented a simplified and faster approach without using Librosa. In a real-time application, at each refresh time, a window of music signals streams in and is first converted from time domain to frequency domain through Fourier Transform. High amplitudes summed across the low frequency band are recognized as candidates for beats, where the low frequency threshold is set to $125Hz$ and the high amplitude threshold is set to 1.5 times the mean amplitudes of all frequency bands. We maintain the running average of inter-beat intervals to help identify the incoming beats among the candidates. From the detected beats, the general (Beats Per Minute) BPM of the music is computed accordingly.

Music Energy Computation. Librosa divides the continuous music input into a sequence of overlap-

ping windows, where each window has 2048 audio samples. Root Mean Square (RMS) value of the amplitudes is computed from the window spectrogram and is defined as the energy E_t of the music window at time t . The sequence of RMS values across all windows constitutes a time-varying representation of the entire music’s energy.

4.3.2 Feature Matcher

Feature matcher receives the computed music features and the \hat{H}_{t+1} embedding predicted from the high-level choreographer as input, and its goal is to find the best dance segment that matches the music.

Cluster Identification. We compute the mean squared distance between each cluster embedding \bar{H}_c and the predicted segment embedding \hat{H}_{t+1} and identified the target cluster with the nearest distance. Cluster identification finds the structurally most probable synonym group of dance segments, where segments perform similar motions in various ways.

Segment Selection. Music BPM helps estimate the preferred frame length of the dance segment, based on which we filtered out too long ($\geq 125\%$) or short ($\leq 70\%$) segments in the nearest cluster, to avoid drastic temporal warping during synchronization. To further identify the best dance segment among the remaining segments in the cluster, we followed the style matching practice proposed in (Aristidou et al., 2017), where music energy computed from audio amplitudes is linked to motion amplitudes features derived from Laban Movement Analysis (LMA) (Laban and Ullmann, 1971). Specifically, eight LMA features consistently related to motion amplitude are computed for each dance segment in the nearest cluster. Grouped by LMA component, i.e. BODY and EFFORT, the eight LMA features $\{f^1 \sim f^8\}$ are listed in Table 1.

The preferred LMA features of the dance segment at time $t + 1$ are computed using the following energy mapping function f_{LMA} :

$$f_{LMA}(E_{t+1}) = \frac{\sum_{i=0}^5 \sum_{j=1}^8 f_{t-i}^j E_{t+1}}{\sum_{i=0}^5 E_{t-i}} \quad (7)$$

f_{LMA} takes the incoming music’s energy E_{t+1} as input and computes the expected LMA features of dance segment at time $t + 1$, based on previous dance segments’ LMA features within the range $[t - 5, t]$ and their corresponding music energy $E_{t-5} \sim E_t$. Dance segment in the cluster with LMA features nearest to the computed $f_{LMA}(E_{t+1})$ is selected, and energetic music input is mapped to the dance segment with a larger range of motions and sharper movements.

Our segment identification process ensures that the selected dance segment is consistent with the music general rhythm and matches the music energy.

Table 1: LMA features associated with motion amplitude.

		Basic Definition	Derived Feature	
		Description	max	mean
BODY	f^1	hands distance		✓
	f^2	left hand-hip distance		✓
	f^3	right hand-hip distance		✓
	f^4	feet distance		✓
EFFORT	f^5	left hand acceleration	✓	
	f^6	right hand acceleration	✓	
	f^7	left foot acceleration	✓	
	f^8	right foot acceleration	✓	

4.3.3 Synchronizer

Dance synchronizer takes the selected dance segment as input, aligns its motion emphasis with the musical beat, and blends the segment smoothly into the long sequence of dance performance.

According to segment phase discussed in 4.1.1, $p = \frac{1}{2}$ corresponds to the motion emphasis of the dance segment; $p \in [0, \frac{1}{2})$ includes the preparation movement reaching to the emphasis; and during $p \in (\frac{1}{2}, 1]$ dance retracts to the end of the segment. Thus we align dance emphasis frame at $T(p = \frac{1}{2})$ with the detected music beat at time T_{beat} . Once aligned, in most cases the preparation phase of the current dance segment overlaps with the retraction phase of its previous segment to a reasonable extent, and we performed spherical linear interpolation (SLERP) on the overlapping part, to splice the current dance segment smoothly with the previous one. In the case of dense music beats, previous segment’s retraction could overlap more than $p = \frac{1}{2}$ with the current segment and cover the current emphasis. Then the current segment is dropped, to avoid modifying dance emphasis and packing too many motions within a short period of time. In the case of disjoint previous retraction and current preparation, we performed SLERP transitioning from the last frame of previous retraction to the beginning of current preparation. The dance synchronization process is illustrated in Figure 5.

5 EXPERIMENTS AND RESULTS

To fully evaluate our two-level approach for dance choreography, we assessed the motion qualities of the synthesized dance under three musical conditions: 1) non-professional singers’ vocalizing through

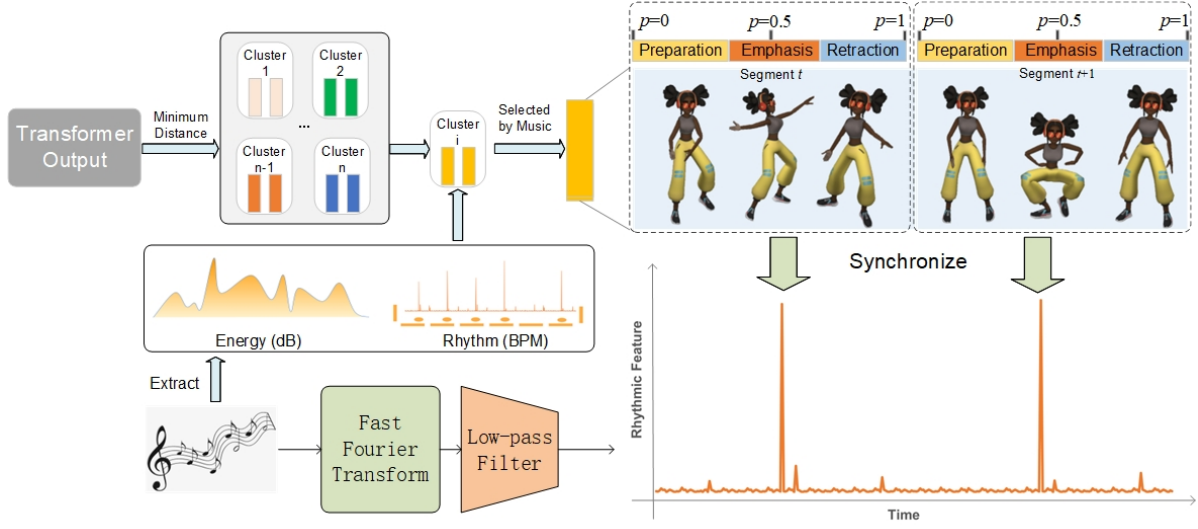


Figure 5: Low-level dance implementer.

a microphone; 2) professional singing with instrumental music playback from the album through a microphone; 3) pure music playback from the album through a microphone. Visual results of choreographed dance motions from our approach are illustrated in the Figure 1 and the supplementary video (<https://youtu.be/PkYDOr-JJDo>). Perceptually our synthesized dance results are not differentiable from the original dance motions in the dataset. To demonstrate the advantages of our two-level approach, we further quantitatively evaluated dance-music synchronization and dance motion diversity of the output motions.

5.1 Quantitative Evaluation

5.1.1 Dance-music Synchronization

We adopted the evaluation practice proposed in (Aristidou et al., 2022) and (Lee et al., 2019) to assess dance-music synchronization. The labelled dance phases were not used when evaluating synchronization, and dance emphasis was derived directly from the result motions using a velocity-based method proposed in (Aristidou et al., 2022), to make fair comparisons with previous work. Music beats were detected based on the method presented in 4.3.1. For each of the three musical conditions, we prepared two 30-second long audio inputs and fed the audio input to our two-level choreographer to synthesize the dance results.

Two dance-music synchronization metrics, **Beat Coverage** and **Beat Hit Rate**, are computed from the music input and the result dance motions. The num-

ber of total musical beats is denoted by B_m , the number of total motion emphasis is denoted by B_k and the number of motion emphasis that are aligned with musical beats is denoted by B_a . According to (Lee et al., 2019), **Beat Coverage** is defined as B_k/B_m , which measures the ratio of motion emphasis to musical beats; **Beat Hit Rate** is defined as B_a/B_k , which is the ratio of aligned motion emphasis to total motion emphasis.

Figure 6 illustrates a section of the choreographed dance sequence given the song "Gangnam Style" as the music input. As proposed in (Aristidou et al., 2022), the local minima in the kinematic velocity are candidates for motion emphasis. The detected music beats and motion emphasis are highlighted in red dashed lines. Table 2 shows the beat score for different music conditions. The average beat coverage and beat hit rate of our approach are 67.3% and 65.3% respectively across all the music conditions, which outperforms baseline sequence model LSTM and previous work Aud-MoCoGAN (Tulyakov et al., 2018), and Dance2Music (Lee et al., 2019).

Table 2: Comparison of beat coverage and beat hit rate.

Method	Beat Coverage	Hit Rate
LSTM	1.4%	51.6%
Aud-MoCoGAN	23.9%	54.8%
Dance2Music	39.4%	65.1%
Ours (professional)	62.1%	66.5%
Ours (non-fessional)	51.3%	64.5%
Ours (pure music)	88.6%	64.8%
Ours (on average)	67.3%	65.3%

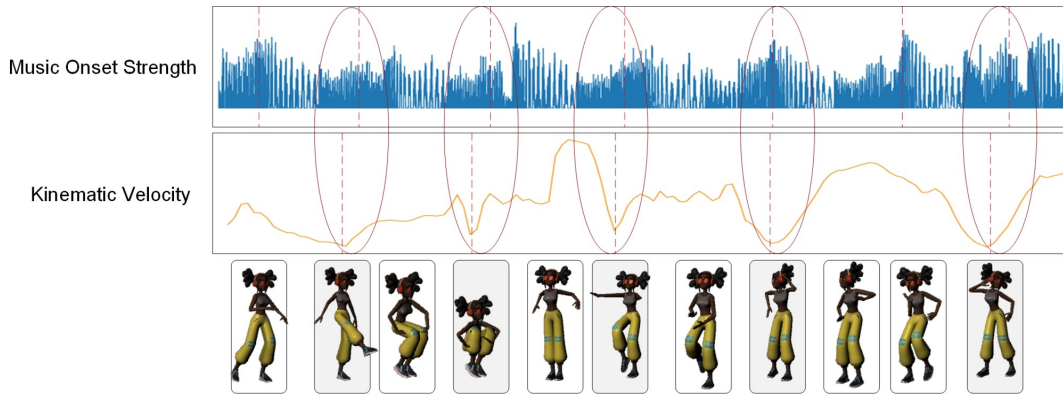


Figure 6: Choreographed dance sequence in Gangnam Style.

5.1.2 Dance Motion Diversity

We evaluated motion diversity of the result dances given the audio input under the three musical conditions that are introduced in 5.1.1. We used the average feature distance similar to (Zhang et al., 2018b) as measurement. The dance hidden features are extracted by the autoencoder presented in 4.1.2, which can better measure the richness and diversity of the motions in the observation space. Diversity results are listed in Table 3. Compared with previous work DeepDance (Sun et al., 2020), Dance2Music (Lee et al., 2019) and ConvSeqGen (Yan et al., 2019), our method achieves the highest diversity score in all musical conditions.

Table 3: Comparison of Motion Diversity.

Method	Motion Diversity
ConvSeqGen	38.7
DeepDance	34.4
Dance2Music	53.2
Ours (professional)	62.9
Ours (non-professional)	59.9
Ours (pure music)	69.1
Ours (on average)	63.9

5.2 Run-Time Performance

We assessed the performance of our two-level approach, and reported the individual run-time speed per dance segment of three major components: 1) high-level choreographer, 2) low-level feature matcher, 3) low-level dance synchronizer. Performance tests are run on a machine with Intel Core i7-12700 Processor 16 GB DDR5. The results are listed in Table 4.

Table 4: Average Run-time Speed per Segment.

Component	Runtime
high-level choreographer	59 ms
low-level feature matcher	14 ms
low-level dance synchronizer	7 ms

5.3 Ablation Study

To better understand the contribution of individual components in our two-level approach, we conducted an ablation study. Each time, we removed one component from the complete workflow and evaluated the system’s performance based on the three critical metrics: **Motion Diversity**, **Beat Coverage**, and **Beat Hit Rate**. From the results listed in Table 5, we can see that the low-level synchronizer primarily ensures alignment between dance emphasis and musical beats, and thus contributes to the high **Beat Hit Rate** of our result motions. Both the high-level choreographer and the low-level feature matcher play pivotal roles in enhancing **Motion Diversity**. The feature matcher is crucial in achieving **Beat Coverage**, as it predominantly selects dance segments that match well with the general rhythm of the music.

6 CONCLUSION AND FUTURE WORK

In this work, we present a novel two-level generation system for choreographing dance motions that are synchronized and compatible with the music input. Our approach decouples the high-level dance structure from its low-level implementation, allowing for the synthesis of expressive dance motions that are both coherent in their genre and varied in their performance. In our results, we have demonstrated the sig-

Table 5: Ablation Analysis on Algorithm’s Performance.

Component (Removed)	Motion Diversity	Beats Coverage	Beat Hit Rate
high-level choreographer	38.7	64.1%	62.5%
low-level feature matcher	34.4	32.1%	61.7%
low-level dance synchronizer	61.3	44.6%	37.4%
Complete Workflow	63.9	67.3%	65.3%

nificance of each component of our system and evaluated the performance of our system in terms of dance-music synchronization and motion diversity.

The present dataset primarily encompasses fast-paced dance, which offers rich movements tied closely to rhythmic patterns. Restricted by the dataset, our current work has limitations in choreographing for relaxing low-rhythmic music. As we continue to evolve our research, we aim to incorporate more dance styles, and adapt our algorithm to synthesize dances for diverse genres of music. This expansion will also enable a more comprehensive understanding of the relationship between dance motions and musical elements across genres. Our current framework mainly focuses on music energy and rhythmic features, reflecting the pulsating beats intrinsic to street dance. However, as we diversify our dance dataset and incorporate other dance forms, we recognize the significance of music spectral features in dance choreography. For example, the tonal quality, timbre, and pitch contours in music can also resonate with different dance styles. Therefore, in the future, in addition to music energy and rhythm, we will investigate more on other spectral features. In our subsequent work, we will also transition our system to operate in real time. Such kind of work will not only promote its applicability but will be helpful for on-device deployment, allowing dancers and choreographers to use our system directly in their practice environments.

ACKNOWLEDGEMENTS

We would like to thank Dr. Scott Bishop for providing constructive feedback to the work, and Yichen Jiang for annotating the dance dataset. Financial support for this work is provided by the McMaster University Startup Fund.

REFERENCES

- Aberman, K., Weng, Y., Lischinski, D., Cohen-Or, D., and Chen, B. (2020). Unpaired motion style transfer from video to animation. *ACM Transactions on Graphics (TOG)*, 39(4):64–1.
- Albrecht, I., Haber, J., and Seidel, H.-P. (2002). Automatic generation of non-verbal facial expressions from speech. In *Advances in Modelling, Animation and Rendering*, pages 283–293. Springer.
- Alemi, O., Françoise, J., and Pasquier, P. (2017). Groovenet: Real-time music-driven dance movement generation using artificial neural networks. *networks*, 8(17):26.
- Alexanderson, S., Nagy, R., Beskow, J., and Henter, G. E. (2023). Listen, denoise, action! audio-driven motion synthesis with diffusion models. *ACM Transactions on Graphics (TOG)*, 42(4):1–20.
- Aristidou, A., Yiannakidis, A., Aberman, K., Cohen-Or, D., Shamir, A., and Chrysanthou, Y. (2021). Rhythm is a dancer: Music-driven motion synthesis with global structure. *arXiv preprint arXiv:2111.12159*.
- Aristidou, A., Yiannakidis, A., Aberman, K., Cohen-Or, D., Shamir, A., and Chrysanthou, Y. (2022). Rhythm is a dancer: Music-driven motion synthesis with global structure. *IEEE Transactions on Visualization and Computer Graphics*.
- Aristidou, A., Zeng, Q., Stavrakis, E., Yin, K., Cohen-Or, D., Chrysanthou, Y., and Chen, B. (2017). Emotion control of unstructured dance movements. In *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation, SCA '17*, pages 9:1–9:10, New York, NY, USA. ACM.
- Chen, K., Tan, Z., Lei, J., Zhang, S.-H., Guo, Y.-C., Zhang, W., and Hu, S.-M. (2021). Choreomaster: choreography-oriented music-driven dance synthesis. *ACM Transactions on Graphics (TOG)*, 40(4):1–13.
- Chuang, E. and Bregler, C. (2005). Mood swings: expressive speech animation. *ACM Transactions on Graphics (TOG)*, 24(2):331–347.
- CMU (2000). Carnegie mellon university mocap database.
- ElKoura, G. and Singh, K. (2003). Handrix: animating the human hand. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 110–119.
- Ellis, D. P. (2007). Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60.
- Englebienne, G., Cootes, T., and Rattray, M. (2007). A probabilistic model for generating realistic lip move-

- ments from speech. *Advances in neural information processing systems*, 20.
- Ferstl, Y., Neff, M., and McDonnell, R. (2019). Multi-objective adversarial gesture generation. In *Motion, Interaction and Games*, pages 1–10.
- Habibie, I., Elgharib, M., Sarkar, K., Abdullah, A., Nyatsanga, S., Neff, M., and Theobalt, C. (2022). A motion matching-based framework for controllable gesture synthesis from speech. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9.
- Hasegawa, D., Kaneko, N., Shirakawa, S., Sakuta, H., and Sumi, K. (2018). Evaluation of speech-to-gesture generation using bi-directional lstm network. In *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, pages 79–86.
- Holden, D., Komura, T., and Saito, J. (2017). Phase-functioned neural networks for character control. *ACM Transactions on Graphics (TOG)*, 36(4):42.
- Holden, D., Saito, J., and Komura, T. (2016). A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 35(4):138.
- Holden, D., Saito, J., Komura, T., and Joyce, T. (2015). Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*, SA '15, pages 18:1–18:4, New York, NY, USA. ACM.
- Iwamoto, N., Kato, T., Shum, H. P., Kakitsuka, R., Hara, K., and Morishima, S. (2017). Dancedj: A 3d dance animation authoring system for live performance. In *International Conference on Advances in Computer Entertainment*, pages 653–670. Springer.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Laban, R. and Ullmann, L. (1971). *The mastery of movement*.
- Lee, H.-Y., Yang, X., Liu, M.-Y., Wang, T.-C., Lu, Y.-D., Yang, M.-H., and Kautz, J. (2019). Dancing to music. *Advances in neural information processing systems*, 32.
- Lee, S., Lee, S., Lee, Y., and Lee, J. (2021). Learning a family of motor skills from a single motion clip. *ACM Transactions on Graphics (TOG)*, 40(4):1–13.
- Levine, S., Krähenbühl, P., Thrun, S., and Koltun, V. (2010). Gesture controllers. In *ACM SIGGRAPH 2010 papers*, pages 1–11.
- Levine, S., Theobalt, C., and Koltun, V. (2009). Real-time prosody-driven synthesis of body language. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–10.
- Li, J., Yin, Y., Chu, H., Zhou, Y., Wang, T., Fidler, S., and Li, H. (2020). Learning to generate diverse dance motions with transformer. *arXiv preprint arXiv:2008.08171*.
- Li, R., Yang, S., Ross, D. A., and Kanazawa, A. (2021). Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13401–13412.
- Li, Z., Zhou, Y., Xiao, S., He, C., and Li, H. (2017). Auto-conditioned LSTM network for extended complex human motion synthesis. *CoRR*, abs/1707.05363.
- Liu, L. and Hodgins, J. (2017). Learning to schedule control fragments for physics-based characters using deep q-learning. *ACM Transactions on Graphics (TOG)*, 36(3):29.
- Liu, L. and Hodgins, J. (2018). Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. *ACM Trans. Graph.*, 37(4):142:1–142:14.
- McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. (2015). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25.
- McNeill, D. (2008). *Gesture and thought*. In *Gesture and Thought*. University of Chicago press.
- Morro Motion (2017). Dance mocap collection. <https://assetstore.unity.com/packages/3d/animations/dance-mocap-collection-102966>.
- Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., and Weber, A. (2007). Documentation mocap database hdm05.
- Neff, M., Kipp, M., Albrecht, I., and Seidel, H.-P. (2008). Gesture modeling and animation based on a probabilistic re-creation of speaker style. *ACM Transactions On Graphics (TOG)*, 27(1):1–24.
- Offi, F., Chaudhry, R., Kurillo, G., Vidal, R., and Bajcsy, R. (2013). Berkeley mhad: A comprehensive multimodal human action database. In *2013 IEEE workshop on applications of computer vision (WACV)*, pages 53–60. IEEE.
- Okamoto, T., Shiratori, T., Kudoh, S., and Ikeuchi, K. (2010). Temporal scaling of leg motion for music feedback system of a dancing humanoid robot. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2256–2263. IEEE.
- Park, J. and Ko, H. (2008). Real-time continuous phoneme recognition system using class-dependent tied-mixture hmm with hbt structure for speech-driven lip-sync. *IEEE Transactions on Multimedia*, 10(7):1299–1306.
- Peng, X. B., Abbeel, P., Levine, S., and van de Panne, M. (2018). Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *CoRR*, abs/1804.02717.
- Peng, X. B., Berseth, G., Yin, K., and Van De Panne, M. (2017). Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 36(4):41.
- Peng, X. B., Ma, Z., Abbeel, P., Levine, S., and Kanazawa, A. (2021). Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (TOG)*, 40(4):1–20.
- Sargin, M. E., Erzin, E., Yemez, Y., Tekalp, A. M., Erdem, A. T., Erdem, C., and Ozkan, M. (2007). Prosody-driven head-gesture animation. In *2007 IEEE International Conference on Acoustics, Speech and Sig-*

- nal Processing-ICASSP'07*, volume 2, pages II-677-IEEE.
- Schulz, A. and Velho, L. (2011). Choreographics: an authoring environment for dance shows. In *ACM SIGGRAPH 2011 Posters*, pages 1-1.
- Shiratori, T., Nakazawa, A., and Ikeuchi, K. (2006). Dancing-to-music character animation. In *Computer Graphics Forum*, volume 25, pages 449-458. Wiley Online Library.
- Shlizerman, E., Dery, L., Schoen, H., and Kemelmacher-Shlizerman, I. (2018). Audio to body dynamics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7574-7583.
- Siyao, L., Yu, W., Gu, T., Lin, C., Wang, Q., Qian, C., Loy, C. C., and Liu, Z. (2022). Bailando: 3d dance generation by actor-critic gpt with choreographic memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11050-11059.
- Smith, H. J., Cao, C., Neff, M., and Wang, Y. (2019). Efficient neural networks for real-time motion style transfer. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2(2):1-17.
- Smith, H. J. and Neff, M. (2017). Understanding the impact of animated gesture performance on personality perceptions. *ACM Transactions on Graphics (TOG)*, 36(4):49.
- Soga, A., Yazaki, Y., Umino, B., and Hirayama, M. (2016). Body-part motion synthesis system for contemporary dance creation. In *ACM SIGGRAPH 2016 Posters*, pages 1-2.
- Sun, G., Wong, Y., Cheng, Z., Kankanhalli, M. S., Geng, W., and Li, X. (2020). Deepdance: music-to-dance motion choreography with adversarial learning. *IEEE Transactions on Multimedia*, 23:497-509.
- Taylor, G. W. and Hinton, G. E. (2009). Factored conditional restricted boltzmann machines for modeling motion style. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 1025-1032, New York, NY, USA. ACM.
- Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. (2018). Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1526-1535.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- Wang, H., Ho, E. S., Shum, H. P., and Zhu, Z. (2019). Spatio-temporal manifold learning for human motions via long-horizon modeling. *IEEE transactions on visualization and computer graphics*, 27(1):216-227.
- Wang, Y. and Neff, M. (2015). Deep signatures for indexing and retrieval in large motion databases. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, MIG '15*, pages 37-45, New York, NY, USA. ACM.
- Wu, N., Green, B., Ben, X., and O'Banion, S. (2020). Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*.
- Xia, S., Wang, C., Chai, J., and Hodgins, J. (2015). Real-time style transfer for unlabeled heterogeneous human motion. *ACM Transactions on Graphics (TOG)*, 34(4):119.
- Yan, S., Li, Z., Xiong, Y., Yan, H., and Lin, D. (2019). Convolutional sequence generation for skeleton-based action synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4394-4402.
- Yoon, Y., Cha, B., Lee, J.-H., Jang, M., Lee, J., Kim, J., and Lee, G. (2020). Speech gesture generation from the trimodal context of text, audio, and speaker identity. *ACM Transactions on Graphics (TOG)*, 39(6):1-16.
- Yumer, M. E. and Mitra, N. J. (2016). Spectral style transfer for human motion between independent actions. *ACM Transactions on Graphics (TOG)*, 35(4):137.
- Zhang, H., Starke, S., Komura, T., and Saito, J. (2018a). Mode-adaptive neural networks for quadruped motion control. *ACM Transactions on Graphics (TOG)*, 37(4):145.
- Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018b). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586-595.
- Zhu, Y., Ramakrishnan, A. S., Hamann, B., and Neff, M. (2013). A system for automatic animation of piano performances. *Computer Animation and Virtual Worlds*, 24(5):445-457.