DanceAnyWay: Synthesizing Beat-Guided 3D Dances with Randomized Temporal Contrastive Learning

Aneesh Bhattacharya[1,2], **Manas Paranjape**[1], **Uttaran Bhattacharya**[3], **Aniket Bera**[1]
[1]Purdue University, USA
[2]IIIT Naya Raipur, India
[3]Adobe Research, USA
{bhatta95, mparanja, aniketbera}@purdue.edu, ubhattac@adobe.com

Abstract

We present DanceAnyWay, a generative learning method to synthesize beat-guided dances of 3D human characters synchronized with music. Our method learns to disentangle the dance movements at the beat frames from the dance movements at all the remaining frames by operating at two hierarchical levels. At the coarser "beat" level, it encodes the rhythm, pitch, and melody information of the input music via dedicated feature representations only at the beat frames. It leverages them to synthesize the beat poses of the target dances using a sequence-to-sequence learning framework. At the finer "repletion" level, our method encodes similar rhythm, pitch, and melody information from all the frames of the input music via dedicated feature representations. It generates the full dance sequences by combining the synthesized beat and repletion poses and enforcing plausibility through an adversarial learning framework. Our training paradigm also enforces fine-grained diversity in the synthesized dances through a randomized temporal contrastive loss, which ensures different segments of the dance sequences have different movements and avoids motion freezing or collapsing to repetitive movements. We evaluate the performance of our approach through extensive experiments on the benchmark AIST++ dataset and observe improvements of about $7\% - 12\%$ in motion quality metrics and $1.5\% - 4\%$ in motion diversity metrics over the current baselines, respectively. We also conducted a user study to evaluate the visual quality of our synthesized dances. We note that, on average, the samples generated by our method were about $9-48\%$ more preferred by the participants and had a $4-27\%$ better five-point Likert-scale score over the best available current baseline in terms of motion quality and synchronization. Our source code and project page are available at https://github.com/aneeshbhattacharya/DanceAnyWay.

Introduction

Dancing is a central human behavior observed across societies and cultures (LaMothe 2019). Being simultaneously a form of expression and communication, the space of dance motions is dense, diverse, and, at the same time, temporally cohesive and structured (Tseng, Castellon, and Liu 2023). The complexity of dance motions and their pervasiveness in our socio-cultural fabric has led to extensive research on generating dancing digital characters for applications such as character design (Mascarenhas et al. 2018), storyboard visualization for consumer media (Kucherenko et al. 2020; Watson et al. 2019), building metaverse tools (Omniverse 2021) and even advancing our understanding of the relationships between music and dance (Brown and Parsons 2008).

Prior methods in dance generation can adapt to different dance genres but may encounter temporal inconsistencies (Fan, Xu, and Geng 2012) and motion freezing and instability (Li et al. 2021a). Using seed poses is a common approach to enforce plausibility (Li et al. 2021a; Zhuang et al. 2022; Li et al. 2020). However, they only provide the initial dance characteristics and become less relevant over time when generating long dance sequences. Diffusion-based approaches (Tseng, Castellon, and Liu 2023) offer better control in the generative process but come at the cost of slow inference speed and heavy parameter tuning for novel datasets. Other approaches tokenize the dance sequences into a finite, learnable set of quantized vectors (Siyao et al. 2022), which can generate long sequences with minimal tuning but tradeoff on the fine-grained diversity of the generated dances.

Different from these approaches, we make two key observations. First, dancers often exhibit bursts or drops of energy at the audio beats. Therefore, the dance steps at the audio beats provide a coarse structure of the dance. Second, we can forego tokenization and explicitly enforce temporal diversity in the generative process to get long dance sequences in the continuous space without freezing or collapsing to repetitive patterns. In this paper, we introduce our method *DanceAnyWay* (Fig. 1), built on these observations, to generate plausible 3D dances from audio. We learn the correlation between the dance motions and the audio at two temporal levels: a coarser *beat level*, which corresponds to the dance poses at the audio beat frames, and a finer *repletion level*, which corresponds to the dance poses at all the other frames. Further, we perform a randomized temporal contrastive loss between segments at the repletion level to enforce diversity of motion between segments that are arbitrarily far from each other. By explicitly learning the correlation between the audio and the dance poses at the beat frames, we can generate plausible beat pose sequences representing the underlying dance characteristics for the entirety of the audio. Given these beat poses, we can generate the remaining or repletion poses while ensuring they are sufficiently diverse.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

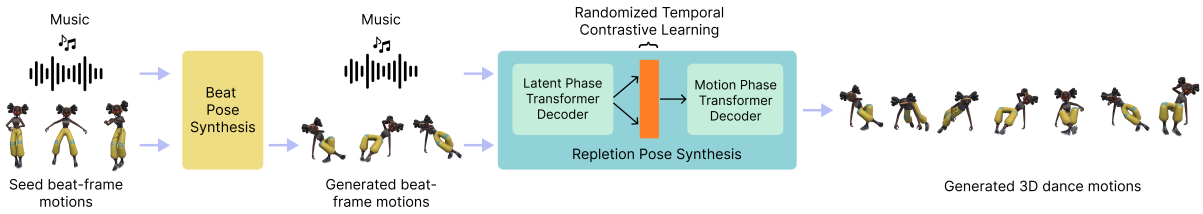

Figure 1: DanceAnyWay: A two-stage hierarchical network that can generate beat-aligned and diverse, fine-grained 3D dances given audio. We render our results with Mixamo characters.

In summary, our main contributions are as follows:

- A temporally hierarchical learning method using sequence-to-sequence and generative adversarial learning to synthesize beat-aligned 3D dance sequences for digital characters synchronized with audio.
- A randomized temporal contrastive loss to generate fine-grained, diverse motions, particularly in the long term.
- Leveraging the spatial-temporal graph representation of the 3D human poses to efficiently learn both the localized (joint-level) and the macroscopic (body-level) movements for different dances.
- An end-to-end pipeline for audio-to-dance generation, which exhibits state-of-the-art performance on multiple quantitative and qualitative evaluations.

Related Work

We briefly review methods for human motion synthesis, particularly from audio inputs such as speech and music.

3D Human Motion-to-Motion Synthesis. 3D motion-to-motion synthesis is richly explored in computer vision and graphics. Classical approaches include kernel-based probability distributions (Galata, Johnson, and Hogg 2001; Pullen and Bregler 2000) to predict the most likely future poses given past poses, and motion graphs (Arikan and Forsyth 2002; Kovar, Gleicher, and Pighin 2008) to represent poses as nodes in a graph and transitioning between those poses according to various linking rules. These models often require significant manual tuning and do not allow for the incorporation of additional input modalities. More recently, learning-based approaches have gained immense traction in this area through convolutional networks (Holden, Saito, and Komura 2016; Holden et al. 2015), recurrent networks (Fragkiadaki et al. 2015; Jain et al. 2015; Ghosh et al. 2017; Bütepage et al. 2017; kuang Chiu et al. 2018; Aksan, Kaufmann, and Hilliges 2019; Du, Vasudevan, and Johnson-Roberson 2019; Wang et al. 2019; Gopalakrishnan et al. 2019), generative adversarial networks (Ruiz, Gall, and Moreno-Noguer 2018), graph convolutional networks (Yan et al. 2019), and transformers (Aksan et al. 2020; Bhattacharya et al. 2021b). Current methods achieve high-quality performance on large-scale datasets and can generate diverse and realistic motions. However, these learning-based approaches are autoregressive and do not condition the motions on additional modalities such as audio.

3D Dance Motion Synthesis from Audio. Procedural methods for audio-to-dance synthesis use approaches such as motion graphs, where the audio rhythms are used to constrain the graph linking rules (Fan, Xu, and Geng 2012; Shiratori, Nakazawa, and Ikeuchi 2006; Pan et al. 2021; Yang et al. 2023; Aristidou et al. 2021; Chen et al. 2021). However, these approaches may suffer from temporal conflicts due to the differences in dance tempos. More recent approaches generate 3D dance motions using deep neural networks. LSTM-based methods (Tang, Jia, and Mao 2018; Yalta et al. 2018) can synthesize long, complex dance sequences by modeling the temporal dependencies in the motion data. GAN-based methods (Lee et al. 2019; Shiratori, Nakazawa, and Ikeuchi 2006) train a generator network to produce realistic dance sequences that match the distribution of a given dataset and a discriminator network to distinguish between the generated and real dance sequences. Transformers-based methods leverage self- and cross-attention mechanisms to capture long-range dependencies between the audio and the dances (Li et al. 2020, 2021a; Tseng, Castellon, and Liu 2023). To overcome the transformers' limitations in generating continuous-space sequences, some transformer variants condense the latent space of dances into a finite set of quantized vectors (Siyao et al. 2022). Other methods segregate the learning into two steps: first generating key poses and then interpolating between them (Li et al. 2022). They have also been paired with diffusion models (Tseng, Castellon, and Liu 2023) to enhance joint-level editing and motion in-betweening capabilities. Large-scale 3D MoCap dance datasets (Alemi, Françoise, and Pasquier 2017; Tang, Jia, and Mao 2018; Zhuang et al. 2022) have played a crucial role in the success of these methods. These datasets have been used to train and test the generative models. Additionally, 3D human models have been mapped to the motion data (Li et al. 2021b), leading to the generation of realistic and expressive dance motions. However, these methods can sometimes result in non-standard poses or regression to mean configurations without exhibiting animated movements due to the high dimensionality of long pose sequences, or cannot adapt to fine-grained dance motions due to quantization. For interpolation-based methods, any error in key pose generation gets propagated to the interpolation network during inference. To overcome these limitations, our method explicitly learns the beat poses to ensure long-term beat alignment and performs a randomized temporal contrastive loss between segments to ensure fine-grained diversity of motions. We also use our generated beat poses only as control signals for interpolation, as a result of which the interpolation process can generate spatially

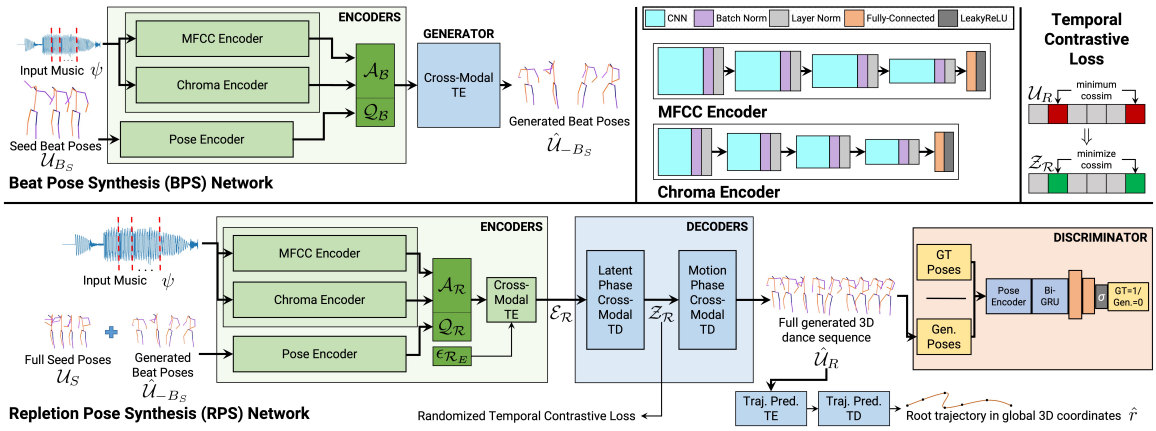

Figure 2: DanceAnyWay Network Architecture. DanceAnyWay consists of two stages, Beat Pose Synthesis (BPS) and Repletion Pose Synthesis (RPS), trained one after the other. BPS *(top row, left)* has a predictor architecture to generate the coarse beat poses, and RPS *(bottom row)* has a generative adversarial architecture to generate all the remaining poses with fine-grained detail, followed by a seq-to-seq trajectory predictor for the global root translations. To train our RPS, we propose an additional randomized temporal contrastive loss *(top row, right)* to enforce motion diversity. For completeness, we also expand our MFCC and Chroma encoders *(top row, middle)*, which have the same architecture but different layer sizes.

and temporally plausible movements regardless of errors in the beat pose generation.

3D Human Motion Synthesis from Other Modalities. Besides music, 3D motions are commonly generated using other modalities, such as speech and text. Co-speech gesture synthesis methods generate accompanying gestures for speech based on learned individual gesticulation patterns. These approaches aim to personalize the synthesis process by capturing the individual characteristics of the speaker (Ginosar et al. 2019), enhance generative capabilities using GAN-based approaches (Ferstl, Neff, and McDonnell 2019), enhance robustness by combining speech, text and speaker identities in the inputs (Yoon et al. 2020), incorporate emotional cues from the audio and gesticulation patterns (Bhattacharya et al. 2021a), explicitly add rhythm-aware information (Ao et al. 2022), and use diffusion models to enable editability (Ao, Zhang, and Liu 2023). In our work, we leverage the audio beat information and the physiological dance movements and use an adversarial framework to improve the plausibility of the generated dances.

Temporally Hierarchical Dance Synthesis

We aim to generate 3D pose sequences for dances given input audio. Our approach is to separately learn the *structure* of the dance described by the *beat poses* or the poses at the beat frames of the audio, and the *finer details* of the dance described by the *repletion poses* or the poses at the remaining frames. To this end, we develop a two-stage learning method consisting of Beat Pose Synthesis (BPS) followed by Repletion Pose Synthesis (RPS). In BPS, given a short sequence of seed beat poses and the audio, we generate the beat poses. In RPS, given all the seed poses, the beat poses following the seed pose duration, and the audio, we generate the remaining poses to complete the dance. Mathematically, we represent the pose at frame t as $\mathcal{U}_t = \left[u_t^{(1)}, \ldots, u_t^{(J-1)}\right] \in \mathbb{R}^{(J-1)\times 3}$, consisting of the unit line vectors denoting the $J-1$ bones corresponding to the J body joints. We take in the audio as a raw waveform and process it into a feature sequence $\mathcal{A} = [a_1, \ldots, a_T] \in \mathbb{R}^{D_\mathcal{A} \times T}$ for some feature dimension $D_\mathcal{A}$ and total temporal length T. We extract the beat frames from the audio using available beat detection methods and represent them as a set $B = \{\text{beat frames in } \mathcal{A}\}$. These beat frames may or may not be equidistant in time. Our BPS takes in the audio features $\mathcal{A}$ and the initial seed beat pose sequence $\mathcal{U}_{B_S} = \{\mathcal{U}_f\}_{f\in B_S}$, where $B_S \subset B$ consists of all the beat frames in B contained within the seed sequence length $T_S \ll T$, and generates the beat poses corresponding to $\mathcal{U}_{-B_S} = \{\mathcal{U}_f\}_{f\in B-B_S}$. Our RPS takes in the audio features $\mathcal{A}$, all the seed poses $\mathcal{U}_S = \{\mathcal{U}_f\}_{f\in\{1,\ldots,T_S\}}$, and the generated beat poses corresponding to $\mathcal{U}_{-B_S} = \{\mathcal{U}_f\}_{f\in -B_S}$, and synthesizes the repletion poses corresponding to $\mathcal{U}_R = \{\mathcal{U}_f\}_{f\in R}$ where $R = \{1,\ldots,T\} - (B \cup S)$. We first fully train our BPS and then use its generated outputs to fully train our RPS, followed by a trajectory predictor for the global root translations. We show the overview of our end-to-end pipeline in Fig. 2 and describe the individual components below.

Beat Pose Synthesis

Our Beat Pose Synthesis (BPS) network takes in the raw audio waveform ψ and the seed beat pose sequence $\mathcal{U}_{B_S}$, and generates the beat poses $\hat{\mathcal{U}}_{-B_S}$. It uses multiple feature encoders to extract rhythmic and semantic information from the audio and the physiological information from the seed beat poses. It combines these features through a transformer-encoder-based generator to synthesize the beat poses.

Feature Encoders. We encode the audio and the seed pose sequences using separate encoder blocks. The *audio en-*

coder block consists of an MFCC encoder and a Chroma encoder. MFCCs naturally capture the human auditory response and are commonly used in tasks such as emotion recognition (Neiberg, Elenius, and Laskowski 2006) and speaker identification (Murty and Yegnanarayana 2006). In our work, we leverage the audio prosody and the vocal intonations (when present) captured by the MFCCs. An MFCC encoder $\mathcal{M}_{\mathcal{B}}$ takes in the MFCCs and their first- and second-order derivatives and uses convolutional layers to learn D_M-dimensional latent feature sequences $\mathcal{A}_{M_{\mathcal{B}}} \in \mathbb{R}^{D_M \times |B|}$ from their localized inter-dependencies as

$$\mathcal{A}_{M_{\mathcal{B}}} = \mathcal{M}_{\mathcal{B}}\left(\text{MFCC}\left(\psi\right); W_{M_{\mathcal{B}}}\right), \tag{1}$$

where $W_{M_{\mathcal{B}}}$ are the trainable parameters. Chroma CENS features capture the melody and pitch in audio, and we use a Chroma encoder $\mathcal{C}_{\mathcal{B}}$ with convolutional layers to transform these Chroma CENS features into $D_{C_{\mathcal{B}}}$-dimensional latent feature sequences $\mathcal{A}_{C_{\mathcal{B}}} \in \mathbb{R}^{D_{C_{\mathcal{B}}} \times |B|}$ based on their localized inter-dependencies as

$$\mathcal{A}_{C_{\mathcal{B}}} = \mathcal{C}_{\mathcal{B}}\left(\text{Chroma}\left(\psi\right); W_{C_{\mathcal{B}}}\right), \tag{2}$$

where $W_{C_{\mathcal{B}}}$ are the trainable parameters. We concatenate these two features into audio features $\mathcal{A}_{\mathcal{B}}$ as

$$\mathcal{A}_{\mathcal{B}} = [\mathcal{A}_{M_{\mathcal{B}}}; \mathcal{A}_{C_{\mathcal{B}}}] \in \mathbb{R}^{D_{\mathcal{A}_{\mathcal{B}}} \times |B|}, \tag{3}$$

where $D_{\mathcal{A}_{\mathcal{B}}} = D_M + D_{C_{\mathcal{B}}}$. For the *pose encoder block*, we adopt the pose encoder architecture of (Bhattacharya et al. 2021a) to learn the physiological variations in the dance motions represented by $\mathcal{U}_{B_S}$. The pose encoder block $\mathcal{P}_{\mathcal{B}}$ outputs latent pose features $\mathcal{Q} \in \mathbb{R}^{D_P \times |B|}$ as

$$\mathcal{Q}_{\mathcal{B}} = \mathcal{P}_{\mathcal{B}}\left(\mathcal{U}_{B_S}; W_{P_{\mathcal{B}}}\right), \tag{4}$$

where $W_{P_{\mathcal{B}}}$ are the trainable parameters.

Transformer-Encoder-Based Generator. We concatenate the latent features $\mathcal{A}_{\mathcal{B}}$ and $\mathcal{Q}_{\mathcal{B}}$ and pass them through a transformer encoder (TE) $\Theta_{\mathcal{B}}$ with cross-attention between the two features to generate $\hat{\mathcal{U}}_{-B_S}$, as

$$\hat{\mathcal{U}}_{-B_S} = \Theta_{\mathcal{B}}\left(\mathcal{A}_{\mathcal{B}} \oplus \mathcal{Q}_{\mathcal{B}}; W_{\Theta_{\mathcal{B}}}\right), \tag{5}$$

where $\oplus$ denotes concatenation and $W_{\Theta_{\mathcal{B}}}$ are the trainable parameters. We note the use of TE here, which generates the entire sequence at once. This is because the beat frames can be irregularly separated in time, and the traditional autoregressive decoder fails to learn these separations.

Repletion Pose Synthesis

In contrast to BPS, our Repletion Pose Synthesis (RPS) network generates a dense sequence of repletion poses, capturing the finer details. Traditional seq-to-seq approaches fail to capture these details and lead to mean regression. To overcome this, we opt for a generative adversarial approach using a generator and a discriminator. The generator takes in the raw audio waveform ψ, the *full* seed pose sequence $\mathcal{U}_S$, all the generated beat poses $\hat{\mathcal{U}}_{-B_S}$ and Gaussian noise $\epsilon_{\mathcal{R}_E}$ for the encoder, and synthesizes the repletion poses $\hat{\mathcal{U}}_R$. The discriminator learns to distinguish between the ground truth and the synthesized pose sequences based on their physiological features, and the generator eventually produces dance motions that the discriminator cannot distinguish from the ground truth, leading to plausible synthesized dances.

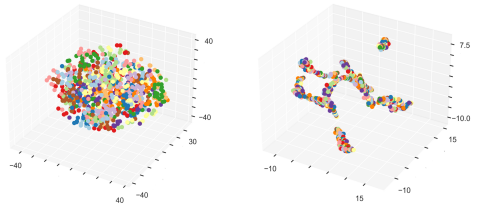

Figure 3: t-SNE Plot of Samples from RPS Latent Decoder Space. Distribution of the features for the m-length segments in $\mathcal{Z}_{\mathcal{R}}$, for 100 random samples (each represented with a different color) in AIST++ (Li et al. 2021b), after training with *(right)* and without *(left)* our RTC loss. Clustering all the sample segments using the RTC loss is necessary to generate diverse motions.

Generator. The RPS generator consists of encoder blocks (similar to the BPS network), followed by a transformer encoder-decoder (TE-TD) architecture with cross-attention. Specifically, we use an MFCC encoder $\mathcal{M}_{\mathcal{R}}$, a Chroma encoder $\mathcal{C}_{\mathcal{R}}$, and a pose encoder $\mathcal{P}_{\mathcal{R}}$, similar in architecture to their BPS counterparts but trained separately with parameters $W_{\mathcal{M}_{\mathcal{R}}}$, $W_{\mathcal{C}_{\mathcal{R}}}$, and $W_{\mathcal{P}_{\mathcal{R}}}$, to obtain the counterpart latent features $\mathcal{A}_{\mathcal{R}}$ and $\mathcal{Q}_{\mathcal{R}}$. Different from BPS, we include $\epsilon_{\mathcal{R}_E}$ as an additional input feature and then use the TE $\Theta_{\mathcal{R}}$ to obtain encoded features $\mathcal{E}_{\mathcal{R}}$, as

$$\mathcal{E}_{\mathcal{R}} = \Theta_{\mathcal{R}}\left(\mathcal{A}_{\mathcal{R}} \oplus \mathcal{Q}_{\mathcal{R}} \oplus \epsilon_{\mathcal{R}_E}; W_{\Theta_{\mathcal{R}}}\right), \tag{6}$$

where $W_{\Theta_{\mathcal{R}}}$ are the trainable parameters. We decode $\mathcal{E}_{\mathcal{R}}$ first into a latent space and then into the motion space, both employing transformer decoders (TDs), as

$$\mathcal{Z}_{\mathcal{R}} = \Phi^{Z}_{\mathcal{R}}\left(\mathcal{E}_{\mathcal{R}}; W_{\Phi^{Z}_{\mathcal{R}}}\right), \tag{7}$$

where $\mathcal{Z}_R = \left\{\mathcal{Z}_f \in \mathbb{R}^{D_Z}\right\}_{f \in R}$ are the D_Z-dimensional latent features and $W_{\Phi^{Z}_{\mathcal{R}}}$ are the trainable parameters. In the subsequent motion decoding phase, we use a TD $\Phi^{U}_{\mathcal{R}}$, as

$$\hat{\mathcal{U}}_{\mathcal{R}} = \Phi^{U}_{\mathcal{R}}\left(\mathcal{Z}_{\mathcal{R}}; W_{\Phi^{U}_{\mathcal{R}}}\right), \tag{8}$$

where $W_{\Phi^{U}_{\mathcal{R}}}$ are the trainable parameters. We perform latent decoding into $\mathcal{Z}_R$, of the same sequence length as $\hat{\mathcal{U}}_{\mathcal{R}}$, to efficiently apply temporal diversity constraints on the smooth, equivalence space of $\mathcal{Z}_R$ rather than on the non-smooth space of unit line vector sequences $\hat{\mathcal{U}}_{\mathcal{R}}$.

Discriminator. Our discriminator takes in 3D dance pose sequences $\widetilde{\mathcal{U}}_{\mathcal{R}}$, which can be either the ground truth $\mathcal{U}_{\mathcal{R}}$ or the generated $\hat{\mathcal{U}}_{\mathcal{R}}$, and uses a pose encoder $\mathcal{P}_{\mathcal{D}}$ (same architecture as $\mathcal{P}_{\mathcal{R}}$) to learn latent pose features $\widetilde{\mathcal{Q}}_{\mathcal{D}} \in \mathbb{R}^{D_P \times |R|}$ based on the physiological variations in the dances, as

$$\widetilde{\mathcal{Q}}_{\mathcal{D}} = \mathcal{P}_{\mathcal{D}}\left(\widetilde{\mathcal{U}}; W_{P_{\mathcal{D}}}\right), \tag{9}$$

Method	Quality		Diversity		BAS	PFC
	FID_k ↓	FID_g ↓	MD_k →	MD_g →	↑	↓
GT	17.10	10.60	10.61	7.48	0.24	0.32
Dnc. Tf.	86.43	43.46	6.85	3.32	0.16	×
DncNet	69.18	25.49	2.86	2.85	0.14	×
DncRev.	73.42	25.92	3.52	4.87	0.19	×
FACT	35.35	22.11	10.85	6.14	0.22	2.25
B'lando	28.16	9.62	7.92	7.72	0.23	1.75
EDGE	20.55	9.49	10.58	7.62	0.27	1.65
DAW	**17.98**	**9.42**	**10.62**	**7.51**	**0.33**	**0.83**
– BPS	18.64	9.73	7.26	4.98	0.22	1.79
– RTC	18.54	9.95	6.91	4.81	0.26	1.05
– rand.	18.82	11.08	7.72	5.28	0.27	2.80

Table 1: Quantitative Evaluation on AIST++ (Li et al. 2021b). Bold = best, underline = best among current methods, × = metric not calculated, arrows are directions for better values: ↑ = higher, ↓ = lower, → = closer to ground truth.

where $W_{P_\mathcal{D}}$ are the trainable parameters. It then uses a bidirectional GRU (BiGRU) of latent dimension $H_\mathcal{D}$ to learn the temporal inter-dependencies in the pose features, followed by a set of FC layers to compress the features into scalar variables and a sigmoid function to compute binary class probabilities $p_\mathcal{D} \in [0, 1]$, as

$$p_\mathcal{D} = \sigma\left(\text{FC}_\mathcal{D}\left(\text{BiGRU}_\mathcal{D}\left(\widetilde{\mathcal{Q}}_\mathcal{D}; W_{L_\mathcal{D}}\right); W_{FC_\mathcal{D}}\right)\right), \quad (10)$$

where we only consider the output of the BiGRU and not its hidden states, $W_{L_\mathcal{D}}$ and $W_{FC_\mathcal{D}}$ denote the trainable parameters, and $\sigma(\cdot)$ denotes the sigmoid function.

Trajectory Predictor. We complete the dances by learning the root trajectory from the generated poses. We use a TE-TD architecture that takes in $\hat{\mathcal{U}}_\mathcal{R}$, learns latent pose features $\mathcal{Q}_\mathcal{T} \in \mathbb{R}^{D_P \times |R|}$ through a TE $\Theta_\mathcal{T}$ with trainable parameters $W_{\Theta_\mathcal{T}}$, and decodes them autoregressively through a TD $\Phi_\mathcal{T}$ with trainable parameters $W_{\Phi_\mathcal{T}}$ to predict the 3D world coordinates of the root, $\hat{r} \in \mathbb{R}^{3\times|R|}$.

Training and Testing

We detail the loss functions we use for training our network, the implementation details, and the testing procedure.

Training Loss Functions

We use pose and leg motion losses to first train our BPS. We then use our proposed randomized temporal contrastive (RTC) loss, pose and leg motion losses, and adversarial losses to train our RPS. We describe our RTC loss below and provide details of the other losses in our appendix.

Randomized Temporal Contrastive (RTC) Loss. While the transformer is currently state-of-the-art for sequence generation (Vaswani et al. 2017), it can lead to freezing and mode collapse for motion sequences such as dances (Li et al. 2021b), where the sequence variables lie in an infinite, continuous space rather than in a finite set of quantized tokens. To address these issues, we consider overlapping segments of length m within each sequence with a sliding length d, and enforce diversity across these segments. We choose a segment n at random and obtain the non-overlapping segment $\bar{n}$ with the minimum cosine similarity to it, as

$$\bar{n} = \arg\min_{x \in N} |\text{cossim}(\mathcal{U}_n, \mathcal{U}_x)|, \quad (11)$$

where N is the set of $\left\lceil \frac{|R|-m}{d} \right\rceil$ segments. We then compute our RTC loss on the RPS latent decoder space (Eqn. 7) as

$$\mathcal{L}_{RTC} = |\text{cossim}(\mathcal{Z}_n, \mathcal{Z}_{\bar{n}})|. \quad (12)$$

This ensures that the segments of our generated sequences are as temporally well-separated as the corresponding training data, enforcing diversity and avoiding freezing or collapse to repetitive motions. The random choice of n is necessary as it prevents the network from memorizing segment positions and makes it focus on *all* the segments across the training epochs in an expected sense. Using the smooth space of the latent decoder sequence $\mathcal{Z}$ instead of the non-smooth motion space of $\hat{\mathcal{U}}$ is also necessary, as it enables stable backpropagations. Our RTC loss thus enables the transformer architecture to operate reliably on continuous sequences. This differs from the commonly used alternative of vector quantization (VQ) followed by tokenized sequence generation (van den Oord, Vinyals, and Kavukcuoglu 2017), which limits the generative power to the finite set of quantized vectors. While some methods improve on the conventional VQ approach by learning separate upper- and lower-body representations (Siyao et al. 2022), their combined representations cannot encompass all possible motions in the full-body motion space.

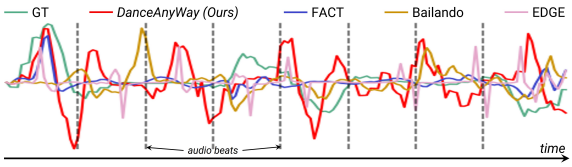

Figure 4: Beat Alignment. Kinetic velocities over time for one ground truth (GT) motion and corresponding generative results. Our method has more peaks and valleys at the beat frames, indicating more alignment with the audio.

Implementation Details

We train our network using 7-second dance clips sampled at 10 fps, *i.e.*, with $T = 70$, and use a seed pose length $T_S = 20$. For our RTC loss, we use segments of length $m = 25$ with sliding window length $d = 5$. We use Librosa (Brian McFee et al. 2015) to extract the MFCC and the Chroma CENS features and compute the beat frames. We use a maximum of $|B| = 20$ beat frames and $|B_S| = 3$ seed beat frames. We use $D_M = 32$, $D_{C_\mathcal{B}} = 4$, $D_{C_\mathcal{R}} = 6$, $D_P = 16$, $\Theta_\mathcal{B}$, $\Theta_\mathcal{R}$, and $\Theta_\mathcal{T}$. $\Theta_\mathcal{B}$ and $\Theta_\mathcal{T}$ have 4 heads and 6 blocks, while $\Theta_\mathcal{R}$ has 8 heads and 6 blocks, $\Phi^Z_\mathcal{R}$ with 3

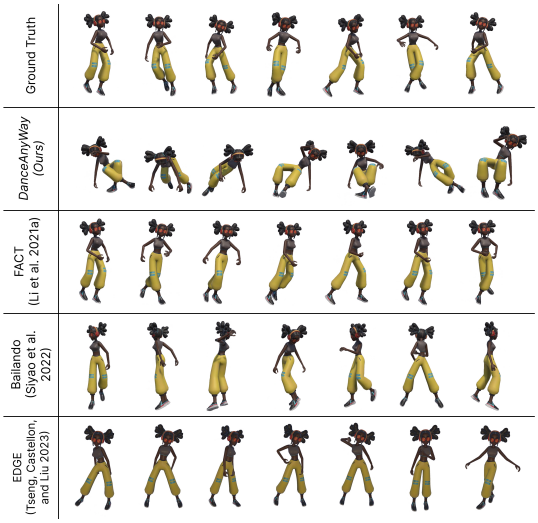

Figure 5: Visualizations on AIST++ (Li et al. 2021b). Sampled frames in a left-to-right sequence for one test sample. Our generated samples are better aligned with beats, more diverse, and have more plausible fine-grained details.

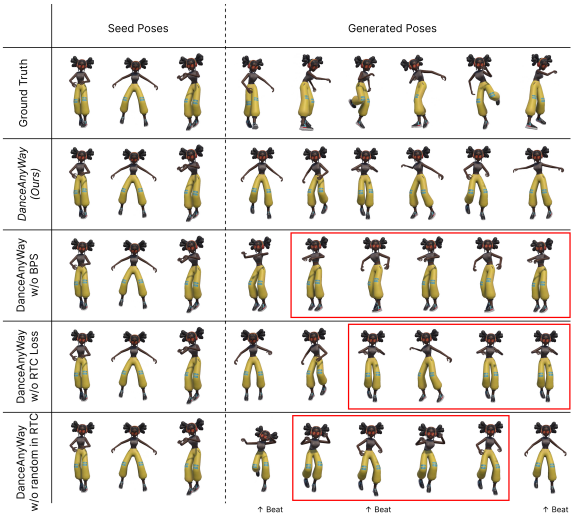

Figure 6: DanceAnyWay Ablations on AIST++ (Li et al. 2021b). Sampled frames in a left-to-right sequence for one test sample. We highlight issues such as misalignment with beats (row 3), lack of motion diversity (row 4), and motion jitter (row 5) with red boxes.

heads and 8 blocks, $\Phi_{\mathcal{R}}^{U}$ with 3 heads and 4 blocks, and $\Phi_{\mathcal{T}}$ with 1 head and 8 blocks. For BPS, we use the Adam optimizer (Kingma and Ba 2014) with $\beta_1 = 0.5$, $\beta_2 = 0.99$, a mini-batch size of 8, a learning rate (LR) of $1e-4$, and train for 500 epochs. For RPS, we use the Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.99$, a mini-batch size of 8, an LR of $1e-4$ for both the generator and discriminator, and train for 250 epochs. For our trajectory predictor, we use the Adam optimizer with $\beta_1 = 0.8$, $\beta_2 = 0.99$, a mini-batch size of 8, a learning rate of $1e-5$, and train for a total of 700 epochs. Training our BPS, RPS, and trajectory predictor takes 6, 16, and 3 hours, respectively, on an NVIDIA A100 GPU.

Inference

During inference, we provide the input audio and the seed poses to our network. BPS generates the beat poses in one prediction step. RPS generates the entire dance and the root trajectories. To render our generated dances on human meshes, we follow the approach of (Li et al. 2021a) to apply them on Mixamo characters (*https://www.mixamo.com*).

Experiments and Results

We describe the benchmark dataset we evaluate on and our quantitative and qualitative performances.

Dataset

We use the benchmark AIST++ dataset (Li et al. 2021b), a large-scale 3D dance dataset of paired music and pose sequences spanning ten dance genres. We use the official dataset splits for training and testing our model.

Evaluation Metrics

We use the following common evaluation metrics (Li et al. 2021a; Siyao et al. 2022; Tseng, Castellon, and Liu 2023).

Fréchet Inception Distance (FID). Following (Li et al. 2021a; Siyao et al. 2022), we compute FID on both the kinetic (k) and the geometric (g) features to measure the generated motion quality relative to the ground truth.

Motion Diversity (MD). Following (Li et al. 2021a; Siyao et al. 2022; Tseng, Castellon, and Liu 2023), we compute MD on both the kinetic (k) and the geometric (g) features as well to measure the diversity of the generated dances relative to the ground truth.

Beat Alignment Score (BAS). BAS measures the temporal alignment of dances with the audio beats. It is essential to understand the rhythmic quality of the dances. We use the BAS implementation of prior work (Li et al. 2021a; Siyao et al. 2022; Tseng, Castellon, and Liu 2023).

Physical Foot Contact Score (PFC). We also report PFC, introduced by (Tseng, Castellon, and Liu 2023) to measure the physical plausibility of the foot movements w.r.t. the ground plane by measuring foot sliding.

Quantitative Evaluations

We compare our proposed method, DanceAnyWay, with the baseline methods of Dance Transformers (Li et al. 2020), DanceNet (Zhuang et al. 2022), DanceRevolution (Huang et al. 2021), FACT (Li et al. 2021a), Bailando (Siyao et al. 2022) and EDGE (Tseng, Castellon, and Liu 2023). We also evaluate three ablated versions of our method: without the BPS network, without the RTC loss, and assigning a fixed "reference" segment instead of using randomization for the RTC loss. We report all the results in Table 1.

Comparison With Baselines. Compared to the best baseline of EDGE (Tseng, Castellon, and Liu 2023), our FID_k

	GT	**DAW**	FACT	Bailando	EDGE
Quality	3.66	**3.58**	2.86	2.23	<u>3.36</u>
Sync	3.61	**3.61**	2.93	2.37	<u>3.60</u>

Table 2: Perceptual Study Scores. Mean preferences on samples generated from AIST++ (Li et al. 2021b) based on the quality of the dances and synchronization with the audio. Bold = best, underline = second-best among all methods.

and FID_g scores are about 12% and 7% better respectively, our MD_k and MD_g scores are about 4% and 1.5% better respectively, and our BAS and PFS improve by about 22% and 50% respectively. We further demonstrate better beat alignment of our generated dances for a test sample in Fig. 4 and visualize snapshots from the generated sequences in Fig. 5.

Comparison With Ablated Version Without BPS. In this ablation, we train only the RPS network with audio and seed poses to synthesize the dance sequences. Without BPS, the RPS network loses alignment with the audio beats as time progresses, leading to lower BAS (Table 1, row 9). These results show the importance of BPS supplying the necessary beat information for long-term motion synthesis.

Comparison With Ablated Version Without RTC Loss. In this ablation, we remove the RTC loss during training. We observe that the motion diversity drops rapidly as time progresses, and the network becomes susceptible to motion freezing and looping over limited motions after a few time steps, leading to lower MD (Table 1, row 10). This also corroborates with how the network learns the RPS latent decoder space, consisting of feature sequences $\mathcal{Z}_{\mathcal{R}}$ (Eqn. 7), with and without the RTC loss (Fig. 3). Using the RTC loss enables the network to avoid mode collapse and enforce diversity by clustering the features at different time steps within each sequence. Without the RTC loss, the network can still generate novel motions in sporadic bursts if it uses BPS. However, the overall motion diversity is limited, as we visualize on a random test sample in Fig. 6, row 4.

Comparison With Ablated Version Without Randomization of RTC Loss. In this ablation, we assign a fixed "reference" segment at the beginning of the sequence and compute the RTC loss w.r.t. this segment. The resultant synthesized dances are unstable as time progresses, changing the joint positions abruptly in an attempt to diversify from the reference segment. Going to the other extreme, minimizing the RTC loss across all segment pairs in each sequence leads to even higher temporal instability. To summarize, the lack of randomization leads to higher FID of the motions (Table 1, row 11) and significant jitter (Fig. 6, row 5).

Perceptual Study

We evaluate the perceived performance of our generated dances through a perceptual study with human participants. For each participant, we select eight random audios from the AIST++ (Li et al. 2021b) test set and generate the corresponding dances using our method and the best-performing baselines methods of FACT (Li et al. 2021a), Bailando (Siyao et al. 2022), and EDGE (Tseng, Castellon, and Liu 2023). We show the participants these generated dances and the corresponding ground truth dances – five dances in total for each audio – in a random order unknown to them. We ask them to rate each dance for each audio on a five-point Likert scale on two aspects: motion quality and synchronization with the audio. To reduce inter-annotator variance, we also provide them guidelines on which aspects of the dances to focus on when assigning the Likert-scale scores. We detail these guidelines in our appendix.

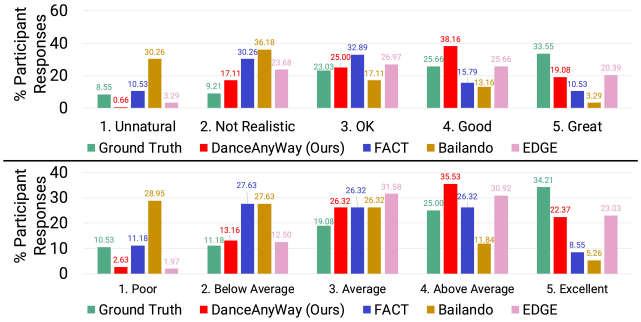

Figure 7: Perceptual Study Response Distributions. Distributions of the Likert-scale scores for all the methods and the ground truth on motion quality *(top)* and synchronization with audio *(bottom)*. We note EDGE and our method with the most responses of 3 or above among the methods.

We report results on 31 responses to our perceptual study, discounting responses that failed our validation checks. 9 identified as female, 20 identified as male, 1 identified as non-binary and 1 did not disclose their gender. 16 participants were between the ages of 18 and 24, 13 between 25 and 35, and 2 above 35. We report the mean Likert-scale scores for the methods and the ground truth in Table 2, and show the distribution of responses in Fig. 7. For motion quality, on average, participants preferred our generated dance motions 14%, 27%, and 4% more compared to FACT, Bailando, and EDGE, respectively. They also marked our generated dances 3 or more in motion quality for 82% of the samples, compared to 59% for FACT, 34% for Bailando, and 73% for EDGE. For synchronization, on average, participants preferred our generated dance motions 14%, 25%, and 0.2% more compared to FACT, Bailando, and EDGE, respectively. They also marked our generated dances 3 or more in synchronization for 84% of the samples, compared to 61% for FACT, 43% for Bailando, and 85% for EDGE.

Conclusion and Future Work

We have presented a novel learning method to synthesize beat-aligned, long-term 3D dances from audio. Through extensive quantitative and qualitative evaluations, we have demonstrated the state-of-the-art performance of our method on a benchmark dance dataset. In the future, we plan to extend our method to explicitly understand different dance styles and make the generation more controllable. We also plan to incorporate dancer-specific capabilities and human-human and human-object interactions.

References

Aksan, E.; Kaufmann, M.; Cao, P.; and Hilliges, O. 2020. A Spatio-temporal Transformer for 3D Human Motion Prediction. *arXiv*.

Aksan, E.; Kaufmann, M.; and Hilliges, O. 2019. Structured Prediction Helps 3D Human Motion Modelling. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 7143–7152.

Alemi, O.; Françoise, J.; and Pasquier, P. 2017. GrooveNet: Real-time music-driven dance movement generation using artificial neural networks. *networks*, 8(17): 26.

Ao, T.; Gao, Q.; Lou, Y.; Chen, B.; and Liu, L. 2022. Rhythmic Gesticulator: Rhythm-Aware Co-Speech Gesture Synthesis with Hierarchical Neural Embeddings. *ACM Trans. Graph.*, 41(6).

Ao, T.; Zhang, Z.; and Liu, L. 2023. GestureDiffuCLIP: Gesture Diffusion Model with CLIP Latents. *ACM Trans. Graph.*

Arikan, O.; and Forsyth, D. 2002. Interactive motion generation from examples. *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*.

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*.

Bhattacharya, U.; Childs, E.; Rewkowski, N.; and Manocha, D. 2021a. Speech2AffectiveGestures: Synthesizing Co-Speech Gestures with Generative Adversarial Affective Expression Learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21. New York, NY, USA: Association for Computing Machinery.

Bhattacharya, U.; Rewkowski, N.; Banerjee, A.; Guhan, P.; Bera, A.; and Manocha, D. 2021b. Text2Gestures: A Transformer-Based Network for Generating Emotive Body Gestures for Virtual Agents. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE.

Brian McFee; Colin Raffel; Dawen Liang; Daniel P.W. Ellis; Matt McVicar; Eric Battenberg; and Oriol Nieto. 2015. librosa: Audio and Music Signal Analysis in Python. In Kathryn Huff; and James Bergstra, eds., *Proceedings of the 14th Python in Science Conference*, 18 – 24.

Brown, S.; and Parsons, L. M. 2008. The Neuroscience of Dance. *Scientific American*, 299(1): 78–83.

Bütepage, J.; Black, M. J.; Kragic, D.; and Kjellström, H. 2017. Deep Representation Learning for Human Motion Prediction and Classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1591–1599.

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.

Du, X.; Vasudevan, R.; and Johnson-Roberson, M. 2019. Bio-LSTM: A Biomechanically Inspired Recurrent Neural Network for 3-D Pedestrian Pose and Gait Prediction. *IEEE Robotics and Automation Letters*, 4(2): 1501–1508.

Fan, R.; Xu, S.; and Geng, W. 2012. Example-Based Automatic Music-Driven Conventional Dance Motion Synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 18(3): 501–515.

Ferstl, Y.; Neff, M.; and McDonnell, R. 2019. Multi-Objective Adversarial Gesture Generation. In *Motion, Interaction and Games*, MIG '19. New York, NY, USA: Association for Computing Machinery. ISBN 9781450369947.

Fragkiadaki, K.; Levine, S.; Felsen, P.; and Malik, J. 2015. Recurrent Network Models for Human Dynamics. *2015 IEEE International Conference on Computer Vision (ICCV)*, 4346–4354.

Galata, A.; Johnson, N.; and Hogg, D. 2001. Learning variable-length Markov models of behavior. *Computer Vision and Image Understanding*, 81(3): 398–413.

Ghosh, P.; Song, J.; Aksan, E.; and Hilliges, O. 2017. Learning Human Motion Models for Long-Term Predictions. In *2017 International Conference on 3D Vision (3DV)*, 458–466.

Ginosar, S.; Bar, A.; Kohavi, G.; Chan, C.; Owens, A.; and Malik, J. 2019. Learning Individual Styles of Conversational Gesture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gopalakrishnan, A.; Mali, A.; Kifer, D.; Giles, L.; and Ororbia, A. G. 2019. A Neural Temporal Model for Human Motion Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Holden, D.; Saito, J.; and Komura, T. 2016. A Deep Learning Framework for Character Motion Synthesis and Editing. *ACM Trans. Graph.*, 35(4).

Holden, D.; Saito, J.; Komura, T.; and Joyce, T. 2015. Learning Motion Manifolds with Convolutional Autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*, SA '15. New York, NY, USA: Association for Computing Machinery. ISBN 9781450339308.

Huang, R.; Hu, H.; Wu, W.; Sawada, K.; Zhang, M.; and Jiang, D. 2021. Dance Revolution: Long-Term Dance Generation with Music via Curriculum Learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jain, A.; Zamir, A. R.; Savarese, S.; and Saxena, A. 2015. Structural-RNN: Deep Learning on Spatio-Temporal Graphs. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5308–5317.

Kingma, D. P.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization.

Kovar, L.; Gleicher, M.; and Pighin, F. 2008. Motion Graphs. In *ACM SIGGRAPH 2008 Classes*, SIGGRAPH '08. New York, NY, USA: Association for Computing Machinery. ISBN 9781450378451.

kuang Chiu, H.; Adeli, E.; Wang, B.; Huang, D.-A.; and Niebles, J. C. 2018. Action-Agnostic Human Pose Forecasting. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1423–1432.

Kucherenko, T.; Jonell, P.; van Waveren, S.; Henter, G. E.; Alexandersson, S.; Leite, I.; and Kjellström, H. 2020. Gesticulator: A Framework for Semantically-Aware Speech-Driven Gesture Generation. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, 242–250. New York, NY, USA: Association for Computing Machinery. ISBN 9781450375818.

LaMothe, K. 2019. The dancing species: how moving together in time helps make us human. *Aeon, June*, 1.

Lee, H.-Y.; Yang, X.; Liu, M.-Y.; Wang, T.-C.; Lu, Y.-D.; Yang, M.-H.; and Kautz, J. 2019. Dancing to Music. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Li, B.; Zhao, Y.; Zhelun, S.; and Sheng, L. 2022. DanceFormer: Music Conditioned 3D Dance Generation with Parametric Motion Transformer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2): 1272–1279.

Li, J.; Yin, Y.; Chu, H.; Zhou, Y.; Wang, T.; Fidler, S.; and Li, H. 2020. Learning to Generate Diverse Dance Motions with Transformer.

Li, R.; Yang, S.; Ross, D. A.; and Kanazawa, A. 2021a. AI Choreographer: Music Conditioned 3D Dance Generation with AIST++. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 13381–13392. Los Alamitos, CA, USA: IEEE Computer Society.

Li, R.; Yang, S.; Ross, D. A.; and Kanazawa, A. 2021b. Learn to Dance with AIST++: Music Conditioned 3D Dance Generation. arXiv:2101.08779.

Mascarenhas, S.; Guimarães, M.; Prada, R.; Dias, J.; Santos, P. A.; Star, K.; Hirsh, B.; Spice, E.; and Kommeren, R. 2018. A Virtual Agent Toolkit for Serious Games Developers. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, 1–7.

Murty, K. S. R.; and Yegnanarayana, B. 2006. Combining evidence from residual phase and MFCC features for speaker recognition. *IEEE Signal Processing Letters*, 13(1): 52–55.

Neiberg, D.; Elenius, K.; and Laskowski, K. 2006. Emotion recognition in spontaneous speech using GMMs. In *Ninth international conference on spoken language processing*.

Omniverse, N. 2021. *NVIDIA Omniverse, https://www.nvidia.com/en-us/omniverse/*.

Pan, J.; Wang, S.; Bai, J.; and Dai, J. 2021. Diverse Dance Synthesis via Keyframes with Transformer Controllers. *Computer Graphics Forum*, 40(7): 71–83.

Pullen, K.; and Bregler, C. 2000. Animating by multi-level sampling. In *Proceedings Computer Animation 2000*, 36–42.

Ruiz, A. H.; Gall, J.; and Moreno-Noguer, F. 2018. Human Motion Prediction via Spatio-Temporal Inpainting. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 7133–7142.

Shiratori, T.; Nakazawa, A.; and Ikeuchi, K. 2006. Dancing-to-Music Character Animation. *Computer Graphics Forum*, 25(3): 449–458.

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 (CVPR)*, 11050–11059.

Tang, T.; Jia, J.; and Mao, H. 2018. Dance with Melody: An LSTM-Autoencoder Approach to Music-Oriented Dance Synthesis. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, 1598–1606. New York, NY, USA: Association for Computing Machinery. ISBN 9781450356657.

Tseng, J.; Castellon, R.; and Liu, K. 2023. EDGE: Editable Dance Generation From Music. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 448–458.

van den Oord, A.; Vinyals, O.; and Kavukcuoglu, K. 2017. Neural Discrete Representation Learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, 6309–6318. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781510860964.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Wang, B.; Adeli, E.; kuang Chiu, H.; Huang, D.-A.; and Niebles, J. C. 2019. Imitation Learning for Human Pose Prediction. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 7123–7132.

Watson, K.; Sohn, S. S.; Schriber, S.; Gross, M.; Muniz, C. M.; and Kapadia, M. 2019. StoryPrint: An Interactive Visualization of Stories. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, 303–311. New York, NY, USA: Association for Computing Machinery. ISBN 9781450362726.

Yalta, N.; Watanabe, S.; Nakadai, K.; and Ogata, T. 2018. Weakly Supervised Deep Recurrent Neural Networks for Basic Dance Step Generation.

Yan, S.; Li, Z.; Xiong, Y.; Yan, H.; and Lin, D. 2019. Convolutional Sequence Generation for Skeleton-Based Action Synthesis. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 4393–4401.

Yang, Z.; Wen, Y.-H.; Chen, S.-Y.; Liu, X.; Gao, Y.; Liu, Y.-J.; Gao, L.; and Fu, H. 2023. Keyframe Control of Music-driven 3D Dance Generation. *IEEE Transactions on Visualization and Computer Graphics*.

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*, 39(6).

Zhuang, W.; Wang, C.; Chai, J.; Wang, Y.; Shao, M.; and Xia, S. 2022. Music2Dance: DanceNet for Music-Driven Dance Generation. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 18(2).