DanceAnyWay: Synthesizing Mixed-Genre 3D Dance Movements Through Beat Disentanglement

Aneesh Bhattacharya
IIIT Naya Raipur
India

Abstract—**We present DanceAnyWay, a hierarchical attention-based generative adversarial learning method to synthesize mixed-genre dance movements 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, and leverages them to synthesize the beat poses of the target dance 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 and couples them with the synthesized beat poses from the coarser level to synthesize the full target dance sequence using an adversarial learning framework. By disentangling the broader dancing styles at the coarser level from the specific dance movements at the finer level, our method can efficiently synthesize dances composed of arbitrarily mixed genres and styles. We evaluate the performance of our method through extensive experiments on both the genre-agnostic TikTok dance dataset and the genre-specific AIST++ dataset and observe improvements over the current baselines. To evaluate the visual quality of our synthesized dances, we also conduct a user-study and observe that participants preferred our synthesized dances over the current baselines.**

I. Introduction

Dance has always served as way to bring humans together; a form of communication the transcends the barriers drawn by language. The evolution of dance brought about different dance forms and various ways for humans to express themselves. The modern digital age enables dance all around to world to be connected, birthing a new age of digital chore graphic dance. Dancing has always been popular on social media platforms and used an open area by amateurs for expressing themselves and as portfolios by aspiring professions wanting to display their skills to attract opportunities [2]. More recently, the growth in popularity of short-form video social media platforms like TikTok have resulted in short-form screen-dancing becoming one of the most popularly consumed content [3]. These dances have their roots mostly in hip-hop but are an amalgamation of several different dance genres. They don't conform to fixed patterns or genres since they are mostly improv dances by amateurs, contrasting them from conventional dances which follow discernible, predictable patterns for separate genres. This randomness makes the dance moves hard to predict and imitate. Computationally, conventional pattern recognition methods perform poorly in this task due to this inherent lack of patterns.

A key way of understanding the flow of movements of this style of dance is to observe the strong correlation between the sporadic bursts or drops of energy in the dances and the beats of the song. Often, knowing the dance-steps during the beats of a song is sufficient enough to guess the entire dance by adding intermediate steps in-between. In this paper, we introduce DanceAnyWay which follows this ideology to generate realistic looking 3D short-form screen-dances. Specifically, given a piece of music and some seed poses, our network is able to generate 10 seconds long dance motions of this style.

The hierarchical design of our network consists of two sub-networks: the beat-pose generator and the pose-repletion network. The beat-pose generator is a sequence to sequence encoder-decoder network. The encoder consists of an audio encoder and a pose encoder. The encoder takes 3 seed beat-poses along with the full music piece which it compresses into a 20 time-step representation to make a coarse representation of the overall song. The decoder uses a sequence to sequence attention network to synthesize 20 dance poses. The target poses for this network to synthesize are the first 20 "beat-poses" of a dance i.e the dance poses occurring on the beats of a song. These generated "beat-poses" along with 2 seconds of input dance poses are then used as a seed input to the pose-repletion network. The pose-repletion network is broadly comprised of two parts: the generator network and the discriminator network where the former is further divided into an encoder-decoder structure. The encoder consists of an audio encoder and a pose encoder. The decoder consists of a pose-decoder and a root-translation-decoder. The audio encoder compresses the audio into a 100 time-step representation, making a more inclusive representation of the input music unlike the former beat-pose audio encoder. Both the networks use an identical pose-encoder which transforms the 3D dance pose sequences into latent affective features using multi scale spatial-temporal graph convolutions (STGCNs). The pose-repletion network outputs a 10 seconds long realistic short-form screen-dance.

Our network architecture outperforms the State-of-the-Art [4] in both genre specific and genre-agnostic 3D dance generation. Our design choices prevent the freezing of synthesized dance poses or their regression to mean poses throughout the timeline of generation as contrasted to prior works
In summary, our contributions are as follows:

- We propose a a hierarchical attention-based generative adversarial learning method to synthesize mixed-genre dance movements of 3D human characters synchronized with music.
- We leverage two different temporal representations of the music and its corresponding dance for network, allowing us to ensure cohesion between the synthesized dance and coarse and fine features of the music.
- We utilize a spatial-temporal graph representation of the 3D human poses to capture the localized and macroscopic body movements throughout the dance and further use it to synthesize dances with greater harmony between the joint movements.
- We provide extensive evaluations and a comprehensive user study to evaluate our performance quantitatively and qualitatively for the generation of realistic short-form screen-dance style 3D dances.

II. Related Works

A. 3D Human Motion Synthesis

Many works have tackled the challenge of synthesizing fluid 3D human motion sequences. Earlier methods used non-linear statistical modeling such as joint probability distributions [64, 10] for motion synthesis, however, lacked motion details. Further research towards motion capture and graphing to provide better frameworks for smooth motion generation through a non-parametric approach. [7, 47] Motion graphing creates a node for each frame in a motion sequence and then collapses the nodes based on similar motions. These collapses create graphs that then through AI planning and graph theory allow for control over synthesizing motion. While this works in theory, there are still challenges in generating realistic transitions between poses that are can be addressed by parameterizing the transition [30]. Advances in deep learning allow for the exploration of neural networks to be trained on large motion capture datasets for producing 3D motion. CNNs have been explored [35, 34] for training datasets based on a motion parameter i.e. the curve a skeleton should follow over a terrain. The result is little for manually pre-processing datasets, however, there is still lack of natural motion being produced without manually kinematics being used to edit the sequences. New implementation of GANs with spatiotemporal tensors have also displayed improvement synthesizing motion. Additional architecture such as RBMs [80], RNNs [24, 88, 12, 87] and Transformers [3, 9] have been explored.

B. Audio To Human Motion Generation

Extensive research has been done in the area of motion synthesis from audio. Much work has been done in producing 2D poses through various implementation of optimization and CNN applications [67, 68, 21]. As a result, the research has produce large datasets for training models for 2D sequence generation based on audio. As 2D sequences fails to provide the level of expressiveness that 3D dances can, new methods have been adapted for 3D dance generation. One method utilizes motion graphing for linking audio to 3D poses for dance generation. Similar to motion synthesis various deep learning architectures such as LSTMs, GANs, and RNNs have been employed to create more advanced models.

C. Music To Dance Generation

Much research related to dance generation had been stagnant due to the lack of available data. Recent work of Li et al. [4] constructed a large refined dataset for dance generation; AIST ++.

III. Approach

Our goal is to generate 3D dance pose sequences for genre agnostic short form screen dances by learning the music-motion correlation. We divide the sequence generation into two sequential parts and solve them hierarchically. The first part deals with generation of 3D dance poses at the beats of a song. This can be formalized as follows: given 3 dances poses $X = \{x_1, x_2, x_3\}$ as seed inputs and the full music sequence, the problem is to generate a sequence of 20 future dance poses $X' = \{x_1, x_2, x_3, ..., x_{20}\}$ corresponding to the first 20 beats of the given music. The second part deals with the generation of the full length 3D dance sequence along with the root translations. This can be formalized as follows: given a 2 second seed sample of 3D dances poses represented by $X = \{x_1, ..., x_T\}$ and the full T' seconds long music sequence, the problem is to generate a sequence of future dance poses $X' = \{x_{T+1}, ..., x_{T'}\}$. The former part is solved by our "Beat-Pose Generator Network" while the latter is solved by our "Pose-Repletion Network".

A. Beat-Pose Generator Network

We propose our Beat-Pose Generator Network for the task of 3D dance pose sequence generation for the beats of a given input music i.e generating 1 dance pose per beat for the first 20 beats of the music. This network is a sequence-to-sequence encode-decoder network which uses a global self-attention mechanism to generate beat-poses. The encoder is comprised of an LSTM encoder which encodes the combined outputs from a separate audio encoder and pose-sequence encoder. The decoder is comprised of an LSTM network with a few Fully-Connected layers to generate beat-poses.

1) Encoder-Decoder Architecture: We encode the audio and seed pose sequence features using separate encoders into a compressed representation of 20 time-steps. For audio encoding, we use an MFCC encoder to encode the combined 2D array representation of the audio MFCC, MFCC Δ and MFCC Δ Δ and a Chroma encoder to encode the Chroma Cens features. For encoding the 3D seed pose features into a latent space, we follow a similar pose-encoding approach as [1]. We consider a directed graph for the pose, where the joints are the vertices, and the edges are directed from the root towards the extremities. We assume the edge lengths are known for each input and represent our 3D poses as directions of the edges. For the 18 joints in our human pose representation, we use the unit vectors of the 17 edges (bones) to represent the body as $U = \{u_1, ..., u_{17}\}$.

The decoder network uses a global self-attention mechanism to generate the beat-poses i.e the unit vectors of each edge of our human pose representation for 20 time-steps, representing the 3D dance poses for the first 20 beats of the song.

2) Beat-Pose Generation: For generation of the beat-poses, we use a global self-attention mechanism at the decoder. We use a linear combination of the attention from the encoder h_e and the decoder h_{d_t} to generate the attention $h_{d_{t+1}}$ for the next time step of the decoder. $h_{d_{t+1}} = 0.4 * h_e + 0.6 * h_{d_t}$ We use a temporal-velocity loss function to train the Beat-Pose Generator model. The target dance poses for this model are the the dance poses at the first 20 beat-frames.

B. Pose-Repletion Network

We propose our Pose-Repletion Network which uses an adversarial learning framework for the task of 3D dance pose sequence generation for the full input music. This network is comprised of a generator and discriminator. The generator comprises of an encoder-decoder architecture which uses a global self-attention mechanism for generating the full 3D dance pose sequence. The discriminator comprises of a pose encoder followed by an LSTM architecture with Fully Connected layers providing a probability vector as an output, determining the decision made by the discriminator in classifying the dance sequence input as real or fake.

1) Generator: The generator encoder follows a similar structure as the Beat-Pose Generator Network's encoder. However, there are 2 key differences between the both:

- The audio and seed pose sequence features are encoded into a compressed representation of a 100 time-steps instead of 20. This change has been done to ensure that during the generation of the full sequence, the Pose-Repletion network has a global view of the finer features of the audio as well as the seed pose sequence.
- The seed pose sequence this network expects is different. The seed pose expected is of the first 2 seconds of 3D dance poses combined with the beat-poses generated by the Beat-Pose Generator. Let $X = \{x_1, ..., x_{t=2}\}$ be the 3D dances poses of the first 2 seconds and $X' = \{x_t, ..., x_{t+n}\}$ represent the beat-poses occurring at times $t >$ 2seconds, then the generator seed pose $X'' = X + \{0, x_1, 0, 0, x_2, ..., x_n, 0, ...\}$ for a total of 100 time-steps where the beat poses are placed at the beat-frames of the full dance sequence.

The generator decoder again follows the similar structure and global self-attention mechanism as the Beat-Pose Generator's decoder with the only difference being that it generates the 3D dance poses in unit vector format $\hat{U} = \hat{u}_1, ..., \hat{u}_{17}$ for a 100 time-steps.

2) Discriminator: Our discriminator takes in the 3D dance pose sequence of length 100 and computes its latent pose feature sequence using the an independent pose encoder with the same architecture as the generator's pose encoder. It then passes this feature sequence through a bidirectional LSTM, and sum the bidirectional outputs to obtain the discriminator embedding sequence. Lastly it transforms the discriminator embedding to a probability vector c ϵ [0, 1] using a set of FC layers such that c $\geq$ 0.5 implies that the discriminator predicts the input 3D dance sequence to be real, and generated otherwise.

3) Pose-Repletion: The idea of pose-repletion is that given the seed input of 2 seconds along with the generated beat poses, the network can fill in the gaps that are present in the generator seed poses i.e generate a complete 3D dance sequence of a 100 time-steps. We have used an adversarial learning mechanism to train the Pose-Repletion Network. We use 2 loss functions: the temporal-velocity loss and the temporal-knee-angle loss along with adversarial loss to ensure that the generate 3D dance sequence are realistic and resemble the short form screen dancing style.

4) Translation Decoder: The translation decoder follows the same overall architecture as the decoder layer of the Pose Repletion Network. However, instead of predicting $\hat{U}$, it predicts the direction vector of translation of the root joint $D = d_z, d_x, d_y$ for each time-step. For predicting the root translation for a generated pose U_t at time T_t, it uses a linear combination of the hidden layer passed into the trained Pose Repletion Network's decoder at T_t combined with its own hidden layer at T_{t-1} in a 4:6 ratio. This network is trained post training of the Pose Repletion Network using a modified temporal-velocity loss.

C. Loss Functions

We have used 4 loss functions: the temporal-velocity loss L_{tv}, the temporal-knee-angle loss L_{tka}, the generative adversarial loss L_{gen} and the discriminator loss L_D

1) Temporal-Velocity Loss: This loss function is a linear combination of a smooth ℓ_1 loss S on the output and target sequences and a smooth ℓ_1 loss on the velocities of the output and target sequences in a 1:1 ratio.

$$S(x, y) = \text{Mean} \begin{cases} \frac{(x_n - y_n)^2}{2\beta} & \text{if} |x_n - y_n| < \beta, \\ |x_n - y_n| - \frac{\beta}{2} & \text{otherwise} \end{cases} \tag{1}$$

$$L_{tv} = S(out, target) + S(out_{velocity}, target_{velocity}) \tag{2}$$

2) Temporal-Knee-Angle Loss: This loss function is a linear combination of a smooth L1 loss on the output and target angles between the femur and shin bone vectors for both legs and a smooth L1 loss on the output and target angular velocity of the angle between the femur and shin bone vectors for both the legs in a 3:7 ratio.

$$L_{tka} = 0.3 * S(out, target) + \\ 0.7 * S(out_{velocity}, target_{velocity})$$

3) Generative Adversarial Loss: It is the loss on the outputs of the discriminator. $L_{gen} = -\epsilon[\log(Disc(\hat{U})]$

4) Discriminator Loss: $L_D = -\epsilon[\log(Disc(U)] - \epsilon[\log(1 - Disc(\hat{U})]$

IV. Experiments and Results

Our network is trained in 2 sequential parts: training the Beat-Pose Generator followed by training the Pose-Repletion Network. For training the Beat-Pose Generator we require 3 seed beat-poses as an input along with audio features extracted from the entire music clip. For the seed beat-poses we use the first 3 beat-frame poses from the target 3D dance sequence. We use Librosa [Librosa] as our audio processing toolbox to extract the music features including: 14-dim MFCCs, 12-dim Chroma and the beat information of the music and further compute the MFCC Δ and MFCC Δ Δ features. The beat-seed poses are converted into the unit vector format and further zero padded along the time-step dimension to a length of 20. We use a batch size of 8 with the Adam [6] optimizer. The learning rate starts from 1e-3 but is dropped to 1e-4 post 700 epochs. The model is trained for a total of a 1000 epochs. The temporal-velocity loss function is used for training.

The Pose-Repletion Network is trained using the outputs obtained from the Beat-Pose Generator Network along with a 2 second long seed-pose sample S_{init} obtained from the target 3D dance sequence. The seed-pose sequence is generated by first zero padding S_{init} to a total length of a 100 time-steps and then placing the generated beat-poses in the appropriate beat-frame positions. The music features required to train this model are the same as the ones used for the Beat-Pose Generator Network. We use a batch size of 8 with the Adam [6] optimizer for both the generator and the discriminator with learning rates of 5e-4 and 5e-6 respectively. The model is trained for 600 epochs. Further, for each forward pass of the network, the generator was optimized 5 times. The loss functions L_G and L_D were used for the generator and discriminator respectively.

$L_G = 500 * (L_{tv} + 0.3 * L_{tka}) + 5 * L_{gen}$

The Translation Decoder is trained post the training of the Pose-Repletion Network with a batch size of 8 and the Adam optimizer [6]. The decoder is trained for a total of a 1000 epochs starting with a learning rate of 5e-4, decaying to 1e-4 post 450 epochs. A modified version of the temporal-velocity loss L_{tvM} is used for training this model. $L_{tvM} = MSE(out, target) + S(out_{velocity}, target_{velocity})$

A. Genre-Specific 3D Dance Synthesis

We have trained our network on the AIST++ dataset [5] which is a large-scale 3D human motion dance dataset with corresponding 3D key points and music for 10 different dance genres. We train and test our model for music clips which are 10 seconds and longer by trimming each music sample to a fixed length of 10 seconds.

B. Genre-Agnostic 3D Dance Synthesis

We have trained our network on the TikTok dataset introduced in [7]. This dataset contained a collection of social media dance videos scraped from the TikTok.

V. Conclusion, Limitations and Future Work

In conclusion we propose a novel 3D dance pose generation technique which outperforms the current SOTA.

References

[1] Uttaran Bhattacharya, Elizabeth Childs, Nicholas Rewkowski, and Dinesh Manocha. Speech2affectivegestures: Synthesiz-ing 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, 2021. Association for Computing Machinery. 2

[2] Makeda Easter. Rise of The Dancefluencer, 2020. 1

[3] Daniel Klug. "it took me almost 30 minutes to practice this". performance and production practices in dance challenge videos on tiktok. 07 2020. 1

[4] R. Li, S. Yang, D. A. Ross, and A. Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 13381–13392, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society 1, 2

[5] Ruilong Li, Shan Yang, David A. Ross, and Angjoo Kanazawa. Learn to dance with aist++: Music conditioned 3d dance generation, 2021 4

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 4

[7] Yasamin Jafarian and Hyun Soo Park. Learning high fidelity depths of dressed humans by watching social media dance videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12753–12762, June 2021 4