DanceFormer: Music Conditioned 3D Dance Generation with Parametric Motion Transformer

Buyu Li,[1] **Yongchi Zhao,**[1] **Zhelun Shi,**[2] **Lu Sheng**[2*]

[1] Huiye Technology,   [2] College of Software, Beihang University
{libuyu, zhaoyongchi}@huiye.tech,   {18373044, lsheng}@buaa.edu.cn

Abstract

Generating 3D dances from music is an emerged research task that benefits a lot of applications in vision and graphics. Previous works treat this task as sequence generation, however, it is challenging to render a music-aligned long-term sequence with high kinematic complexity and coherent movements. In this paper, we reformulate it by a two-stage process, *i.e.*, a key pose generation and then an in-between parametric motion curve prediction, where the key poses are easier to be synchronized with the music beats and the parametric curves can be efficiently regressed to render fluent rhythm-aligned movements. We named the proposed method as Dance-Former, which includes two cascading kinematics-enhanced transformer-guided networks (called DanTrans) that tackle each stage, respectively. Furthermore, we propose a large-scale music conditioned 3D dance dataset, called Phantom-Dance, that is accurately labeled by experienced animators rather than reconstruction or motion capture. This dataset also encodes dances as key poses and parametric motion curves apart from pose sequences, thus benefiting the training of our DanceFormer. Extensive experiments demonstrate that the proposed method, even trained by existing datasets, can generate fluent, performative, and music-matched 3D dances that surpass previous works quantitatively and qualitatively. Moreover, the proposed DanceFormer, together with the PhantomDance dataset, are seamlessly compatible with industrial animation software, thus facilitating the adaptation for various downstream applications.

1   Introduction

Automatically generating music conditioned 3D dances is an appealing but challenging task emerged in the research community of vision and graphics, which can significantly benefit various downstream applications in AR/VR, games, films and even the social networks. Beyond human actions like walking, jumping and sitting with atomic functionalities, dancing is a type of artistic performances that focuses more on its choreography, *i.e.*, sequential steps and movements with high kinematic complexity that are synchronized with the beats and rhythms of the music. It is challenging for humans with professional training to generate expressive

---

*Lu Sheng is the corresponding author.

choreographies, no matter how hard a machine is to generate visually plausible dances accompanied with the music.

Most of the prior works (Lee et al. 2019; Li et al. 2021; Lee, Kim, and Lee 2018; Tang, Jia, and Mao 2018; Li et al. 2020; Zhuang et al. 2020; Alemi, Françoise, and Pasquier 2017) formulated the music conditioned 3D dance generation as a sequence generation problem, where each frame in the sequence describes the human pose by the joint-level rotations and translations. However, rendering a long-term pose sequence with high kinematic complexity and coherent movements is still an open question.

In this paper, we would like to exploit a popular animation strategy (Lasseter 1987; Thomas, Johnston, and Thomas 1995) in the field of computer graphics, where the motion of characters is efficiently rendered by interpolating poses in keyframes through parametric curves. This coarse-to-fine strategy is especially beneficial in our task since it both ensures kinematic complexity and motion conherency in the generated dances, where the poses in the keyframes can be generated and synchronized with the music beats due to their co-occurrence (Lee et al. 2019; Li et al. 2021), and the parameteric curves can be efficiently regressed so as to produce diverse motion patterns that are consistent with the rhythms of the music. It also allows user-controlled temporal resolutions, and generates smooth sequences with fewer temporal flickers (Williams 2012). Moreover, formulating dances as a sequence of key poses and parametric motion curves would be seamlessly compatible with industrial animation pipelines. Therefore, we would like to tackle the music conditioned 3D dance generation from this new perspective, and decompose this task into two easier sub-tasks, namely, the key pose generation and the subsequent parametric motion curve regression.

To this end, we propose DanceFormer, a two-stage music conditioned 3D dance generation framework that consists of two cascaded Transformer-like (Vaswani et al. 2017) networks (called DanTrans), where the first one is to generate key poses and the second one is to regress the parameters of the motion curves between adjacent key poses. At first, we employ an off-the-shelf beat track algorithm (Ellis 2007) to detect beats from the input music (*e.g.*, around 30 seconds). The DanTrans for the key pose generation takes a sequence of music spectrum features centered at each beat as the input, and then generate a sequence of key poses that

synchronizes with the sequence of the music beats. And the DanTrans for the parametric motion curve regression employs a different sequence of music spectrum features that are extracted between adjacent beats, and feeds the generated key poses as well to regress the sequence of the parameters for each motion curve. In our implementation, we employ the Kochanek-Bartels splines (Kochanek and Bartels 1984) to model motion curves, which benfits for rendering various movements from a small set of control parameters.

Each DanTrans contains a *Wave Encoder* as a stack of transformer encoders to extract audio features, and a *Motion Decoder* constructed by a stack of transformer decoders to auto-regressively predict key poses or parameters of the motion curves. To be specific, the Motion Decoder contains a new Kinematic Propagation Module (KPM) and a Structured Multi-head Attention Module (SMAM) to enhance the spatial correlation for the dance generation process based on the Kinematic topological relations of the human body. This motion decoder increases the physical significance and thus alleviates unreasonable predictions seldom appeared in real dances. These DanTrans networks are adversarially trained in accompany with ℓ_2 reconstruction losses.

In order to train the proposed DanceFormer, we also propose a new PhantomDance dataset that is produced by professional animators rather than reconstruction from 2D videos (Lee et al. 2019; Li et al. 2021; Lee, Kim, and Lee 2018) or by motion capture (Zhuang et al. 2020; Tang, Jia, and Mao 2018). The reconstructed data can not ensure the accuracy due to the limitation of recent 3D reconstruction algorithms. The motion capture methods are more accurate but still suffer from noise, such as limb flickers and joint mismatches. In contrast, our PhantomDance dataset is produced by a team of experienced animators instructed by professional dancers. The animated dances are encoded by poses from keyframes and motion curve parameters, which thus can produce more fluent and expressive human pose sequences that matches the input musics.

Extensive experiments are conducted on PhantomDance and AIST++ (Li et al. 2021), demonstrating that Dance-Former attains state-of-the-art results and significantly surpasses other works both qualitatively and quantitatively.

Above all, the contributions are summarized as follows:

(1) A new perspective to model the music conditioned 3D dance generation as key pose generation and parametric motion curve regression, which follows the animation principles and simultaneously ensures kinematic complexity and motion coherency.

(2) A novel framework named DanceFormer that is consists of two Transformer-like networks called DanTrans, where the first network generates key poses and the second one regresses the parameters of the motion curves. More importantly, the DanTrans networks calculates attentions in adaptive temporal ranges and explicitly enhances the kinematic correlation among the outputs.

(3) The PhantomDance dataset, as the first dance dataset crafted by professional animators instead of 3D reconstruction or motion capture, provides more smooth and expressive dances that are synchronized with music, and directly compatible with industrial animation pipelines.

2   Related Works

3D Dance Synthesis   3D dance synthesis has been addressed in a number of ways. The traditional methods were based on retrieval from the dance dataset (Takano, Yamane, and Nakamura 2010; Chao et al. 2004). Recently, deep learning models have been widely employed in this task, attaining visually pleasing results, such as those using convolutional neural networks (Lee, Kim, and Lee 2018; Zhuang et al. 2020), recurrent neural networks (Tang, Jia, and Mao 2018), or using variational auto-encoders (Lee et al. 2019) to enhance the diversity of the generated dances. CSGN (Yan et al. 2019) generates the dance sequences by enhancing the skeleton-level relations with a graph neural network (GNN), but the proposed Kinematic Propagation Module (KPM) used in our DanceFormer adopts forward and inverse kinematic message passing to better involve the kinematic correlations. ChoreoMaster (Chen et al. 2021) introduces a choreography-oriented choreomusical embedding framework, and use the embedding to retrieve dance snippets in the motion graph for the input music. Because it doesn't generate new dances, we don't compare our method with it. As the Transformer (Vaswani et al. 2017) achieves a great success in sequence-to-sequence generation tasks in NLPs, some of the most recent works on the dance generation also bring it into use. Li et al. (2020) presented a two-stream motion transformer generative model. AI Choreographer (Li et al. 2021) presents a cross-model transformer with future-N supervision for auto-regressive motion prediction. These works directly generate the sequence of human poses, while the proposed DanceFormer generates a series of key poses and accompanied parameters of the motion curves, which ensures necessary kinematic complexity, motion coherency, adaptive temporal resolutions in the generated dances, and moreover, compatible with the industrial animation software, such as Maya, Unity and *etc*.

3D Dance Dataset   The widely used motion datasets, such as Human3.6M (Ionescu et al. 2013) and AMASS (Mahmood et al. 2019), collected common actions like walking, running, jumping and sitting. However, these datasets are hard to adapt for the dance generation task due to the huge distribution gap between daily motions and the dance movements, as well as the absence of carefully aligned music-dance pairs. In fact, high-quality 3D dance movements with synchronized musics are extremely hard to collect. Many existing datasets are limited by quantity or quality of data. Some of them just use sequences of 2D keypoints to represent dances (Lee et al. 2019; Lee, Kim, and Lee 2018). AIST++ (Li et al. 2021), as the largest dataset up to date, presented a 5-hours 3D dance set. But it gathered dances by reconstructing 3D poses from 2D multi-view videos, thus the accuracy of pose parameters may not guaranteed. The other works use motion capture to build dataset (Alemi, Françoise, and Pasquier 2017; Tang, Jia, and Mao 2018; Zhuang et al. 2020), in which the pose reliability is better but misalignment between the dance-music pair is inevitable. On the contrary, the proposed PhantomDance is the largest public 3D dance dataset up to date and its data quality is much higher than existing datasets thanks to careful labels from experi-

enced animators with the help of professional dancers. The stored key poses and parametric motion curves also benefit for animations by industrial softwares.

3 The PhantomDance Dataset

3.1 Data Collection

We collect 260 popular dance videos from more than 13 genres from over 100 different subjects (dancers) on YouTube, NicoNico and Bilibili, which have 9.6 hours in total. Then a team of experienced animators produced the 3D dance animations, as sequences of key poses and parametric motion curves. An expert dancer provided professional instructions to teach the animators, thus improve the holistic expressiveness and detail richness of the animated dances, as well as the synchronization with beats and rhythms of the accompanied musics. This dataset is named PhantomDance, and we will make it publicly available to facilitate future research.

3.2 Data Formatting

In our PhantomDance dataset, the dances can be animated in industrial animation software, such as Maya (Derakhshani 2012), on a standard SMPL (Loper et al. 2015) character model with 24 body joints. We followed Human3.6M (Ionescu et al. 2013) and provided a subject-invariant 3D human skeleton representation for all the 260 motion files, with unified limb lengths. The motion is parameterized as the parametric curves of root (the hip joint) position and the rotations of the 24 skeleton joints. For each sequence we take the root position on the first frame as the the origin of the 3D coordinate. And the rotations are expressed with quaternions. That is, there are $3 + 4 \times 24 = 99$ curves in each motion file. Each parametric motion curve is represented as the Kochanek-Bartels Spline interpolated from labeled key poses synchronized with extracted music beats, where curve segment between adjacent two key poses can be controlled by a fixed set of parameters.

We use the parametric motion curves to represent dance movements because (1) its formulation is analytical and can be densely sampled with various temporal resolutions, which is very useful for real-time rendering; (2) its shape is in nature continuous and thus consistent with the aesthetic principles when evaluating dances.

3.3 In Comparison to Previous Datasets

Table 1 shows the comparison between our PhantomDance dataset with the other public music-conditioned 3D Dance Datasets. The collected dances in PhantomDance mainly cover 13 genres, ranging from Urban Dance to Chinese Classical Dance, with extra genres from animes or idol groups which are hard to classify. The beats per minute (BPM) of these dances range from 75 to 178. Apart from labeling the detailed 3D positions and rotations of each pose, our dataset also provides the ground-truth music beats and parameters of each motion curve, thus directly facilitates animation or model learning that is compatible with industrial software.

Besides the aspect of dataset scale, the PhantomDance dataset also has high quality. Motion data in the other

Dataset	Pos/Rot	Beat	Curve	Genres	Music	Seconds
GrooveNet	✓/✗	✗	✗	1	3	1380
EA-MUD	✓/✗	✗	✗	6	23	1849
Dance w/ Melody	✓/✗	✗	✗	4	61	5640
AIST++	✓/✓	✗	✗	10	60	18694
PhantomDance	✓/✓	✓	✓	13+	260	34667

Table 1: Our PhantomDance dataset *v.s.* the other datasets, namely GrooveNet (Alemi, Françoise, and Pasquier 2017), EA-MUD (Sun et al. 2020), Dance with Melody (Tang, Jia, and Mao 2018) and AIST++ (Li et al. 2021).

datasets are collected from optical motion capture system (Alemi, Françoise, and Pasquier 2017; Tang, Jia, and Mao 2018; Sun et al. 2020) or 3D reconstruction from multi-view videos (Li et al. 2021). Motion capture systems can not totally avoid data noise like limb flicker and joint mismatch. And video based 3D reconstruction suffers from more serious noise like foot slippery and limb twist due to the limitation of reconstruction algorithms. Thus professional animators are needed for data correction. In contrast, motion data in PhantomDance are produced by expert animators from scratch with the guidance of a professional dancer. And we have a strict quality checking process so that it costs about 15 months to finish the dance labeling. More details of our PhantomDance dataset and the qualitative comparison with the other datasets can be seen in our project page[1].

4 DanceFormer

As illustrated in Figure 1, we propose the DanceFormer, a two-stage music conditioned 3D dance generation framework that consists of two cascaded transformer-like networks. Each network is named as DanTrans, where the first one generates key poses, and the second one predicts the parameters of the motion curves between adjacent key poses. Each DanTrans network contains a wave encoder that is similar as the standard transformer encoder (Vaswani et al. 2017), and a motion decoder with a Kinematic Propagation Module (KPM) and a Structured Multi-head Attention Module (SMAM), which enhances the kinematic correlation for the dance generation process based on the Kinematic topological relations of the human body. Adversarial learning is applied during the training in both stages.

4.1 Stage 1: Key Pose Generation

Following the rhythm (beats) is a basic principle in dance theory (Goodridge 1999), and previous studies also show that music beats and kinematic inflexions have strong consistency in time (Lee et al. 2019; Li et al. 2021). Thus we define the key poses directly as the poses on the beats, which is represented as position and rotation (in quaternion) of each skeleton joint, with the dimension of $(3+4) \times 24 = 168$. We first leverage the a dynamic programming algorithm (Ellis 2007) to track the time of each beat in the music. Our target in stage 1 is to generate a sequence of key poses to match these extracted beats.

The input sequential music features are music spectrum features between a window centered at each beat. We use a

---

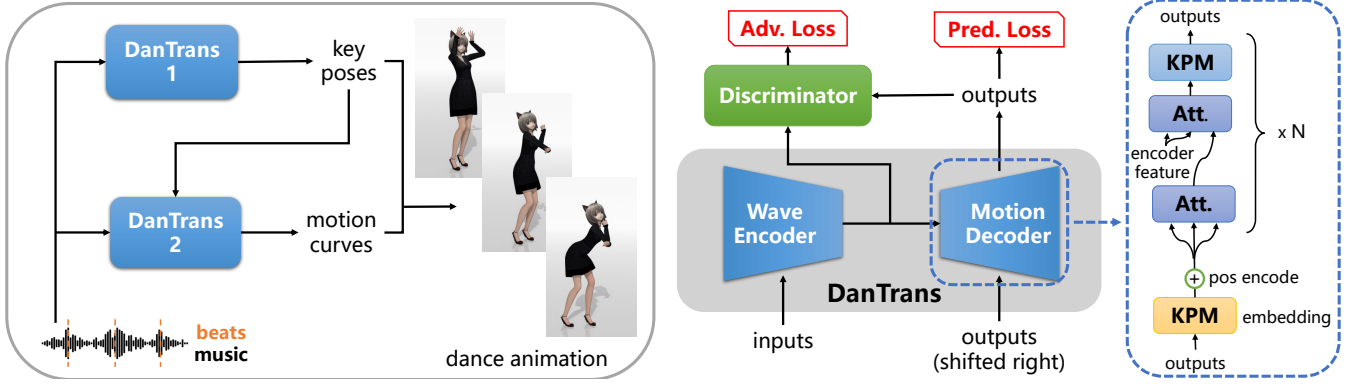[1]https://huiye-tech.github.io/post/danceformer/

Figure 1: **Overview of the workflow of DanceFormer.** The networks in stage 1 take as input the music wave features in the neighborhood of each beat and output a sequence of key poses. Then the networks in stage 2 utilize the generated key poses and the music wave features between each two key poses to predict the motion curve in between. Each network adopts the proposed transformer-like network, called DanTrans. The DanTrans has a new transformer decoder, which uses the proposed Kinematic Propagation Module (KPM). Note that the attention module (Att. in the figure) is a structured multi-head attention.
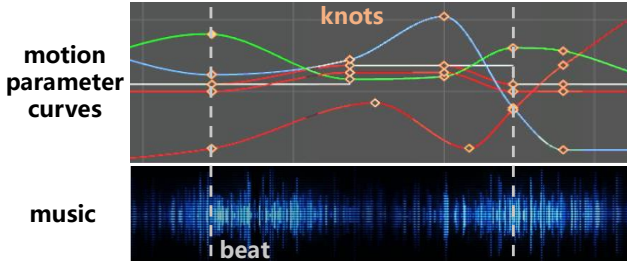

Figure 2: **Motion Curves.** The figure shows an example of 4-knot TCB spline parameterization for motion curves.

window of size 0.8 seconds in our implementation since the lowest beat per-minus (BPM) is 75 in common dance musics (which is validated in our PhantomDance dataset, and also valid in other datasets). We choose the Hamming window to tackle the window overlap in music with higher BPMs. In each window, we calculate the 40-dimensional form of MFCC (Logan et al. 2000) that has 13 coefficients of Mel filters cepstrum coefficients with the first and second order differences (13×2 dimensions) and the energy sum (1 dimension). We further append a 12-dimension chroma feature to the feature of every beat and finally attain the inputs to the wave encoder. And the motion decoder combines the encoded features with the right shifted outputs and sequentially predict the key poses, as indicated by the sequence-to-sequence prediction strategy based on the standard transformer (Vaswani et al. 2017).

4.2 Stage 2: Parametric Motion Curve Prediction

Based on the generated key poses, the DanTrans in stage 2 is targeted to generate the motion curves in between.

The motion curves are defined as the values of transformation parameters with regard to time. The transformation parameters are translation (x, y, z) and rotation (r_x, r_y, r_z, r_w) in quaternion of each joint. We use a multi-knots Kochanek-Bartels splines (Kochanek and Bartels 1984) to model each curve. It is a sort of cubic Her-

mite spline that is used in animation editing softwares. The Kochanek-Bartels spline is also named TCB spline because it has 3 parameters: t for tension, b for bias and c for continuity. While for a multi-knot TCB spline, we should also determine the intermediate knots and their tangents. Figure 2 shows a example of motion curve parameterization.

In our implementation, we use one 4-knot TCB spline to fit one motion curve between two adjacent key poses, as an optimal tradeoff between fitting accuracy and representation compactness. Since the endpoints in the curve are just on the key poses, we have 7 control parameters to predict, namely the t, c, b and the position of the two intermediate knots.

Similar to the DanTrans in stage 1, we extract the sequence of the MFCC features from the music between every two adjacent beats as the input for the wave encoder, and the motion decoder auto-regressively outputs the 7 control parameters that are aligned with each MFCC feature. Note that the motion decoder requires the right shifted output sequences as its input, which should be additionally concatenated with the two key poses at the beginning and the end of the curve, which are generated in stage 1.

4.3 Network Structure of DanTrans

DanTrans has a similar Network structure to the standard transformer (Vaswani et al. 2017) as illustrated on the right of Figure 1, except that the motion decoder employs the proposed Kinematic Propagation Module (KPM) and the Structured Multi-head Attention Module (SMAM) in each transformer decoder layer. We employ a stack of $N = 6$ transformer decoders in the motion decoder of each DanTrans. Note that an additional KPM is applied at the beginning of the motion decoder for output embedding.

Kinematic Propagation Module The Kinematic Propagation Module (KPM) are constructed on the basis of human body structure, *i.e.*, the 24 skeleton joint nodes with a tree topology. The networks consist of a stack of so-called FK-IK block as illustrated in Figure 3. Note that the fully connected layer (FC) represents the complete BN-Linear-ReLU
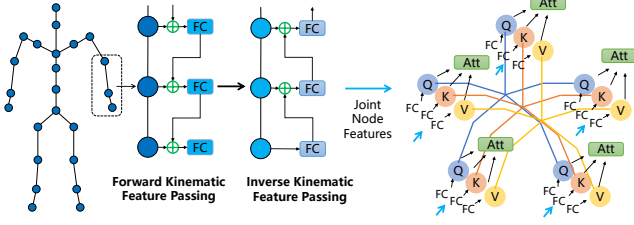
Figure 3: **The Kninematic Propagation Module (KPM) and the Structured Multi-head Attention.** Features are embedded and passed along the kinematic chains forward and inversely with fusion operations. Note that the endpoints of feature passing are root and leaf nodes, and we just show a part of them for clearer view.

module, while we use FC to represent it for simplicity.

The FK-IK block has two steps, *i.e.*, the forward kinematic (FK) feature passing and then the inverse kinematic (IK) feature passing, respectively. The forward kinematic feature passing step firstly performs a feature embedding on the root node and pass it to the neighbor child nodes along the kinematic chain. After receiving the features from the parent node, the child node performs feature fusion and then conducts feature embedding by linear projection. The FC layers are utilized for feature embedding and the fusion adopts the addition operation. The backward kinematic feature passing step has a similar feature embedding and the passing process is performed from the leaf nodes to the root. These procedures are similar to the operations of forward kinematics (FK) and inverse kinematics (IK) in computer graphics and robotics. The FK-IK block attempts to encode features into the motion controlling parameter space and to bring physics constraints into the model.

For the output embedding process at the beginning of the motion decoder, we use a KPM with one FK-IK block. And for feed-forward process in the intermediate layers, we adopt a stack of 2 FK-IK blocks for each KPM. All the KPMs have the same output feature size of 64.

Structured Multi-Head Attention The features outputted by KPM are arranged as the joint nodes, they naturally act as heads in the multi-head attention mechanism used in common transformers (Vaswani et al. 2017). In the multi-head cross-attention, the encoded features from the input music are processed by two sets of 24 paralleled FC layers with the output size of 64, which then act as the values and keys. Note that no concatenation is needed after attention calculation since the features should maintain the body structure, and directly fed into the succeeding KPM module. Thus this process saves computation compared to the typical multi-head attention that concatenates features from each head and uses a big linear layer for feature embedding. The structured multi-head attention module, also shown in the right of Figure 3, defines the head as the node in the body structure, which has explicit physical significance. To some degree this design is more reasonable than the multi-heads in the original transformer.

4.4 Training Objective

We adopt an adversarial training strategy that regards each DanTrans network as an individual generator, and learns two discriminators to judge whether distributions of the generated sequences and the groundtruth ones are aligned or not. Each discriminator consists of two modules to process the music feature and the output/groundtruth key pose sequence (in stage 1) or the parameters of the motion curves (in stage 2), and the processed features are combined and fed into the binary classifier to predict the authenticity of the outputs conditioned on the input music. All modules in the discriminator are 3-layer MLPs. Moreover, when updating the generator, we also include an ℓ_2 reconstruction loss to avoid unwanted solutions that are far away with real data.

5 Experiments

5.1 Implementation Details

Since most musics in PhantomDance have the verse-chorus form, we divide the 260 dance animations into 1000 sequences. Among them 900 pieces of music-dance pairs are used for model training, and the other 100 are split into the test set. We carefully pick the test set to ensure it cover 13 genres and the BPM range from 80 to 180. We follow the official training/testing splits in AIST++ (Li et al. 2021). To gather the ground-truth labels for the training of the DanceFormer, we use the beat track algorithm (Ellis 2007) to extract beats and fit the parameter motion curves from the provided pose sequences. The DanceFormer is end-to-end trained using 4 TITAN Xp GPUs with a batch size of 8 on each GPU. We use the Adam optimizer with betas $\{0.5, 0.999\}$ and a learning rate of 0.0002. The learning rate drops to 2e−5, 2e−6 after 100k, 200k steps. The model is trained with 300k steps for *AIST++* and 400k steps for PhantomDance. The dimension of features in DanceFormer is 256 unless otherwise specified.

5.2 Evaluation Metrics

Normalized Power Spectrum Simularity (NPSS) It is an evaluation metric for long-term motion synthesis compared to Mean Square Error (MSE). We just follow its official implementation (Gopalakrishnan et al. 2019) and compute NPSS in the joint motion space $\mathbb{R}^{T \times N \times 7}$ (4 for joint rotation represented as quaternion and 3 for joint position).

Frechet Distance (FD) It is proposed by AIST++ (Li et al. 2021) that has two metrics, namely PFD for position and VFD for velocity. We also employ it to calculate the distribution distance of joints.

Position Variance (PVar) It evaluates the diversity of the generated dance. In AIST++, a piece of music corresponds to more than one dance, but only one in PhantomDance. So we make a modification to the metric **PVar** in the experiments on PhantomDance where we compute it along different music pieces which have identical length.

Beat Consistency Score (BC) It is a metric for motion-music correlation. We follow (Li et al. 2021) to define kinematic beats as the local minima of the kinetic velocity. BC

Component	NPSS ↓	PFD ↓	VFD ↑	PVar ↑	BC ↑
DanceFormer	**8.03**	**114.03**	**0.55**	**0.912**	**0.785**
Curve to Frames	14.55	1132.4	1.06	0.233	0.317
KPM to Linear	10.54	162.44	0.79	0.592	0.637
KPM to GNN	10.34	146.43	0.69	0.647	0.692

Table 2: **Ablation study about our DanceFormer on the PhantomDance dataset.** ↓ means that lower results indicate better methods, and ↑ *vice versa.*

Att Algorithm	NPSS ↓	PFD ↓	VFD ↑	PVar ↑	BC ↑
Global	9.73	151.53	0.72	0.649	0.737
Local	9.01	125.59	0.68	0.744	0.763
Gaussian Local	8.93	117.14	0.64	0.753	0.764
Learned Local	**8.03**	**114.03**	**0.55**	**0.912**	**0.785**

Table 3: **The comparison of different Attention mechanisms.** ↓ means that lower results indicate better methods, and ↑ *vice versa.*

computes the average distance between every music beat and its nearest kinematic beat with the following equation:

$$BC = \frac{1}{|\mathcal{B}^x|} \sum_{i=1}^{|\mathcal{B}^x|} \exp \left(-\frac{\min_{\forall t_j^x \in \mathcal{B}^x} ||t_j^x - t_i^y||^2}{2\sigma^2} \right) \quad (1)$$

where $\mathcal{B}^x = \{t_j^x\}, \mathcal{B}^y = \{t_i^y\}$ are kinematic beats and music beats respectively and $\sigma = 3$. Note that BC has a similar form to Beat Alignment Score (BA) proposed in (Li et al. 2021), but they are different in essence. BA forces every kinematic beat to match a music beat, but a dance usually has many small kinematic beats that occur between music beats. Moreover, a music synchronized dance just need to ensure the most salient music beats are accompanied with the action emphasis (kinematic beats). So our proposed BC, which finds matched kinematic beat for each music beat, is more appropriate in this case.

5.3 Ablation Study

To validate the effectiveness of our DanceFormer, we conduct the ablation study on our PhantomDance dataset.

Effectiveness of Curve Prediction To evaluate the effectiveness of our curve prediction, we sample 60-FPS poses from the data and use a standard transformer to directly predict the sequence. The input sequences are extracted from a 0.5s sliding window with a step of $1/60$ second on the music waves. Its result is the second line of Table 2 as "Curve to Frame". The experimental result demonstrates the advantage of our pose-to-curve two stage generation framework.

Effectiveness of KPM To study the impact of the KPM component, we first replace it with the standard feed-forward (linear layers) modules in vanilla transformer (Vaswani et al. 2017). And the multi-head attention also comes back to the common form. The third line in Table 2 shows that the KPM has significant improvement in motion generation quality. To further study the FK-IK processing, we introduce a comparison architecture which substitutes the FK-IK block with a 2-layer graph neural network (GNN) (Scarselli et al. 2008). This GNN fuses the joint features according to the adjacency matrix of undirected graph

defined by the joint structure. The experimental result shows that GNN has better performances compared to the baseline ("KPM to Linear"), which proves that introducing spatial correlations benefits the representation of the learning model. While the proposed KPM surpasses the trivial GNN by a large margin, validating that involving spatial correlation and physics constraints into the network brings in significant improvements in the generation quality that is evaluated by NPSS, PFD and VFD.

Variants of Attention In our implementation, we find that local attention (Luong, Pham, and Manning 2015) is more proper for the motion sequence generation task than the global attention in standard transformers. It can be explained by the temporal locality of motions. That is, the pose several seconds ago has no direct influence on the current pose. Moreover relationship between motion state is stronger with the time going closer. So we further compare several attention algorithms and the results are shown in Table 3. We use a sequence length of 17 in the simple local attention experiment shown on the second line. And we add a Gaussian mask on the attention results before `softmax` with a standard variance of 4. The result is on the third line and it has a slight improvement. Finally we use a learned mask to be optimized with the networks and obtain the best result. The trained mask has a triangle shape on the whole with some humps. Our DanceFormer employs this learnable local attention in our implementation.

5.4 Comparison with Other Methods

We mainly compare our method with AI Choreographer (Li et al. 2021), which to our knowledge obtains the state-of-the-art results for music-conditioned dance generation. The other two most related works, namely Li et al. (Li et al. 2020) and Music2Dance (Zhuang et al. 2020) are also compared. The experimental comparisons are performed on both the AIST++ and our PhantomDance datasets.

Quantitative Comparisons The results are viewed in Table 4. Our method outperforms Li et al. (Li et al. 2020) and Music2Dance (Zhuang et al. 2020) by a considerably large margin. And it also surpasses AI Choreographer (Li et al. 2021) significantly in the quality related metrics (a 30% gain of NPSS, a 28% gain of PFD and a 27% gain of VFD), diversity metrics (a 46% gain of PVar) and beat consistency metrics (a 93% gain of BC). The promising results on the dance quality metrics are mainly owing to the network structure of KPM that enhances the kinematic correlations. The generation diversity is ensured by the adversarial learning scheme. Note that the upper bound of the metric BC is 1, which means there exactly exists a kinematic beat at the time of each music beat. So our results have relatively high beat consistency score, which is due to the proposed two-stage framework of our DanceFormer. Since no metric emphasizes the fluency of the dance performance, the advantages of motion curve formulation can only be well revealed in qualitative results, as shown in Figure 4.

Qualitative Results and User Study Figure 4 provides a sequence of frames about the generated dances by our

Figure 4: **Visualization of generated results of DanceFormer.**

Method	AIST++					PhantomDance				
	NPSS ↓	PFD ↓	VFD ↑	PVar ↑	BC ↑	NPSS ↓	PFD ↓	VFD ↑	PVar ↑	BC ↑
Li et al. (Li et al. 2020)	16.31	5595.91	3.40	0.019	0.359	18.34	7944.78	5.84	0.014	0.175
Music2Dance (Zhuang et al. 2020)	14.74	2367.26	1.13	0.215	0.378	15.94	3147.89	3.74	0.267	0.223
AI Choreographer (Li et al. 2021)	8.29	113.56	0.45	0.509	0.452	10.62	164.33	0.73	0.624	0.388
DanceFormer	**6.01**	**84.32**	**0.34**	**0.734**	**0.782**	**8.03**	**114.03**	**0.55**	**0.912**	**0.785**

Table 4: **Dance generation evaluation on the AIST++ and the PhantomDance datasets.** Our method outperforms other baselines in terms of quality, diversity and beat consistency. Especially, due to our two-stage prediction schema, our model has superior performance in term of BC indicating that our model can generate dances which better match the given music. ↓ means that lower results indicate better methods, and ↑ *vice versa*.
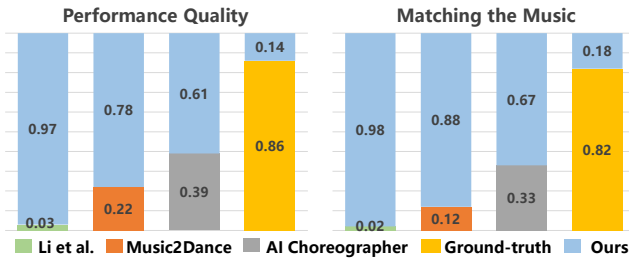

Figure 5: **Results of the user study.** We conduct a user study to ask participants to choose the better dances from pairwise comparisons. The criteria includes the performance quality and matching the music. The number denotes the percentage of preference on the comparison pairs.

DanceFormer. These results show that the proposed method can provide diverse movements with high kinematic complexity. Video results accompanied with musics, including the comparisons with the other works can be found in the project page[2].

We also conducted a user study to evaluate the quality of the generated music-conditioned dances. All the 100 sequences of the validation set of PhantomDance were used for the study. And then we collect the generated dances by our method and the compared baselines (Li et al. 2021, 2020; Zhuang et al. 2020). In addition, the ground-truth dances are also included. The user study was conducted using a pairwise comparison scheme. For each of the 100 musics, we provide 4 pairs in which our results occur with the

results from the baseline methods or the ground truth. Thus 400 pairs were provided to the participants, and they were asked to make two choices for each pair: "Which dance is a better performance (more fluent, graceful and pleasing)?" and "Which dance matches the music better?". There are 100 participants in the user study. Figure 5 shows the user study results, where our DanceFormer outperforms the other methods on both criteria. Most of the participants rated that our method generates better dances in performance quality compared with other works, and even more participants held the opinion that the dances generated by our model better match the musics.

6 Conclusion

In this work, we propose a new perspective to model the music-conditioned 3D dance generation task. Different from previous works that define the outputs as sequences of poses, we formulate them as key poses and in-between motion curves. The curve representation makes the generated results more fluent and graceful. Based on this formulation, we propose the transformer-based DanceFormer with the novel DanTrans architecture consisting of the KPM module for better modeling kinematic correlations. DanceFormer thus yields high-quality results in the experimental comparisons. Moreover, we propose the PhantomDance Dataset, the first music-conditioned 3D dance dataset that uses curves to represent body motion, and it is the largest 3D dance dataset with the best visual quality up to date.

[2]https://huiye-tech.github.io/post/danceformer/

References

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

Chao, S.-P.; Chiu, C.-Y.; Chao, J.-H.; Yang, S.-N.; and Lin, T.-K. 2004. Motion retrieval and its application to motion synthesis. In *24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings.*, 254–259. IEEE.

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.

Derakhshani, D. 2012. *Introducing Autodesk Maya 2013*. John Wiley & Sons.

Ellis, D. P. 2007. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1): 51–60.

Goodridge, J. 1999. *Rhythm and timing of movement in performance: Drama, dance and ceremony*. Jessica Kingsley Publishers.

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*, 12116–12125.

Ionescu, C.; Papava, D.; Olaru, V.; and Sminchisescu, C. 2013. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7): 1325–1339.

Kochanek, D. H.; and Bartels, R. H. 1984. Interpolating splines with local tension, continuity, and bias control. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, 33–41.

Lasseter, J. 1987. Principles of traditional animation applied to 3D computer animation. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 35–44.

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, J.; Kim, S.; and Lee, K. 2018. Listen to dance: Music-driven choreography generation using autoregressive encoder-decoder network. *arXiv preprint arXiv:1811.00818*.

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 *The IEEE International Conference on Computer Vision (ICCV)*.

Logan, B.; et al. 2000. Mel frequency cepstral coefficients for music modeling. In *Ismir*, volume 270, 1–11. Citeseer.

Loper, M.; Mahmood, N.; Romero, J.; Pons-Moll, G.; and Black, M. J. 2015. SMPL: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6): 1–16.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*.

Mahmood, N.; Ghorbani, N.; Troje, N. F.; Pons-Moll, G.; and Black, M. J. 2019. AMASS: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5442–5451.

Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; and Monfardini, G. 2008. The graph neural network model. *IEEE transactions on neural networks*, 20(1): 61–80.

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.

Takano, W.; Yamane, K.; and Nakamura, Y. 2010. Retrieval and Generation of Human Motions Based on Associative Model between Motion Symbols and Motion Labels. *Journal of the Robotics Society of Japan*, 28(6): 723–734.

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*, 1598–1606.

Thomas, F.; Johnston, O.; and Thomas, F. 1995. *The illusion of life: Disney animation*. Hyperion New York.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Williams, R. 2012. *The animator's survival kit: a manual of methods, principles and formulas for classical, computer, games, stop motion and internet animators*. Macmillan.

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*, 4394–4402.

Zhuang, W.; Wang, C.; Xia, S.; Chai, J.; and Wang, Y. 2020. Music2dance: Music-driven dance generation using wavenet. *arXiv preprint arXiv:2002.03761*.