You Never Stop Dancing: Non-freezing Dance Generation via Bank-constrained Manifold Projection

Anonymous Author(s)
Affiliation
Address
email

Abstract

One of the most overlooked challenges in dance generation is that the auto-regressive frameworks are prone to freezing motions due to noise accumulation. In this paper, we present two modules that can be plugged into the existing models to enable them to generate non-freezing and high fidelity dances. Since the high-dimensional motion data are easily swamped by noise, we propose to learn a low-dimensional manifold representation by *an auto-encoder with a bank of latent codes*, which can be used to reduce the noises in the predicted motions, thus preventing from freezing. We further extend the bank to provide explicit priors about the future motions to disambiguate motion prediction, which helps the predictors to generate motions with larger magnitude and higher fidelity than possible before. Extensive experiments on AIST++, a public large-scale 3D dance motion benchmark, demonstrate that our method notably outperforms the baselines in terms of quality, diversity and time length.

1 Introduction

Dancing to music has been one of the most popular art forms since ancient days. It can vividly express humans emotions and fulfill social communications even before symbolic languages came along. Nowadays, there are a booming number of people sharing their dance videos on the popular media platforms such as YouTube and TikTok, which drives the strong need for automatic AI choreography to help users create their own dances. The task is related to general human motion prediction [19, 22, 27, 34, 35, 37] except that it poses new challenges: i) dance generation needs to produce high-fidelity motions for about three minutes to cover a music which is much longer than that in general motion prediction; ii) dance generation needs to handle more diverse and stylistic motions (*e.g.*, ballet Jazz, hip-pop, *etc.*). The high spatio-temporal complexity requires more expressive models in order to generate high-fidelity motions.

The state-of-the-art methods follow a cross-modal prediction framework where the future motions are predicted based on the past motions and music in an auto-regressive way [10,14,18,26]. However, the generated motions are prone to freezing and small-magnitude motions after only several seconds. The main reason is that the prediction error will accumulate in the generation process, and eventually can not be handled by the predictors. Besides, motion prediction suffers from huge uncertainty and ambiguity because of the high spatio-temporal complexity of the task. As a result, the models tend to predict mean poses [37] instead of large-magnitude and high-fidelity motions if there are not informative priors about the future motions.

In this paper, we present two modules that can be plugged into the existing auto-regressive models to achieve non-freezing large-magnitude motion generation. Figure 1 shows an overview. Firstly, to prevent from error accumulation, we present *RefineBank* to learn a low-dimensional manifold representation for the high-dimensional motion data. It equips an auto-encoder with a bank of

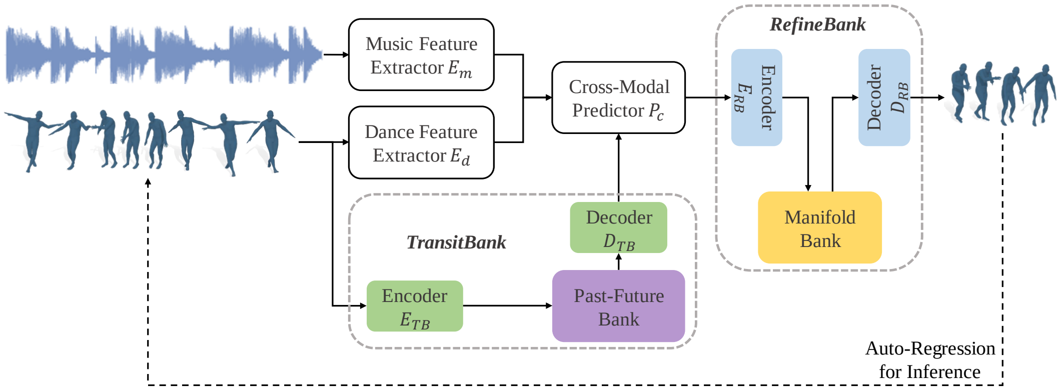


Figure 1: The motion prediction framework. It first uses the past motions to query TransitBank to retrieve the priors about the future motions, which are fed to the predictor to get high-fidelity motion predictions. Then RefineBank refines the predictions to reduce error accumulation.

latent codes to *tightly* constrain the manifold to be close to the ground-truth (GT) motions and meanwhile far from the ones with noises. This representation allows us to remove the noises in the predicted motions by projecting them to the learned manifold as illustrated in Figure 2. With RefineBank, the baseline method [18] can already generate full choreography for complete musics in the dataset without freezing. Secondly, inspired by the fact that most dances can be coarsely constructed by a number of basic motion segments, we present *TransitBank* to learn and memorize the frequently used <past, future> motion dynamics on top of the manifold. Given past motions, it provides explicit priors about the future motions which help reduce the uncertainty and ambiguity in motion prediction. It can effectively facilitate the higher-fidelity dance motion generation with larger magnitude than possible before.

We evaluate our approach on the AIST++ dataset [18]. Not only does our method notably outperform the baselines on the existing metrics, but also shows better results on our newly introduced freezing rate metric. In addition, the user study indicates that people have obvious preferences toward the dances generated by our method. Our contributions are summarized as follows: 1) This is the first time we see evidence that the prediction-based methods can generate long-term dance motions on AIST++, which paves the way for full choreography for entire music; 2) We present *RefineBank* and *TransitBank* which can be plugged into most motion-prediction based methods to achieve long-term non-freezing dance generation; 3) We introduce new metrics that can quantitatively evaluate the freezing situations in the motion sequences.

2 Related Work

Prediction-based Methods This line of work treats dance generation as a motion prediction problem and has achieved promising results. A number of network architectures have been proposed including CNNs [8, 14], RNNs [1, 10, 28, 31, 36], GCNs [4, 23, 32], GAN [26] and Transformers [11, 16–18]. Several works focus on the alignment of motion and music. For instance, Sun et al. [26] and Li et al. [16] use a classifier to test the authenticity of the predicted motion conditioned on the music. Zhang et al. [36] and Huang et al. [11] learn the dance style embeddings to provide prior information to the predictor. Few works have studied the freezing problem. Huang et al. [10] propose a curriculum learning strategy to bridge the gap between training and inference by alternately feeding predicted and ground-truth motions to the predictor. Li et al. [18] present future-n full-attention to replace the traditional shift-by-1 casual-attention to leverage the temporal context. However, the predicted motions still freeze after several seconds.

Retrieval-based Methods Some earlier works compose a complete dance by retrieval [12, 21, 24, 25]. They select the closest predefined motion segments in a pre-built database based on music, and construct a sequence with the proper transition routines. Lee et al. [15] and Ye et al. [33] use deep networks to generate future motion segments from the input music and past motion segments. Duan et al. [3] propose an attention-based MLP to translate all music phrases to motion segments. Chen

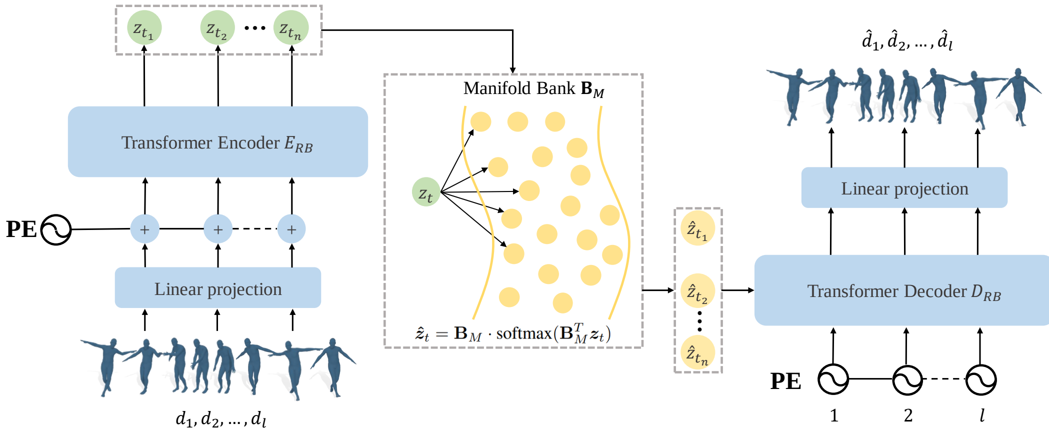


Figure 2: **RefineBank**: It takes a motion sequence with l frames as input, projects them to a low-dimensional manifold, and reconstructs the sequence with minimum noises.

et al. [2] propose to generate dance sequences through traversing the node transition routines on a motion graph and introduce the choreography-oriented constraints. However, they cannot generate new motions beyond the database. Our approach is a prediction-based method so it enjoys all of the benefits such as being able to generate new motions beyond the database. But different from the previous methods, we present two plug-in modules that allow longer-term motion generation. Our approach is also inspired by the retrieval-based methods in that we construct a bank/database to learn common motion dynamics for the predictor. The difference is that the bank is automatically learned from data without human efforts.

3 Preliminaries

Given the past and future music features $\boldsymbol{m}_{1:t} \parallel \boldsymbol{m}_{t+1:t+K}$, and the past dance motions $\hat{\boldsymbol{d}}_{1:t}$, we aim to predict the future motions $\hat{\boldsymbol{d}}_{t+1:t+K}$. The input music features $\boldsymbol{m}_t \in \mathbb{R}^{35}$ are obtained from Librosa [20] and the motions are represented by SMPL parameters. Since shapes are dance-irrelevant, we only use the pose parameters and global translation vector in SMPL for $\hat{\boldsymbol{d}}_t \in \mathbb{R}^{219}$. We follow the state-of-the-art dance generation architecture [18] as shown in Figure 1. It has a music feature extractor E_m, a dance feature extractor E_d, and a cross-modal predictor P_c. The auto-regressive prediction process can be mathematically expressed as:

$$\hat{\boldsymbol{d}}_{t+1:t+K} = P_c(E_m(\boldsymbol{m}_{1:t+K}), E_d(\hat{\boldsymbol{d}}_{1:t})). \tag{1}$$

Although previous works such as [10, 18] have attempted to improve the long-term prediction performance, the predictions may still freeze after only several seconds.

4 Method

The dance motions are believed to lie on a low-dimensional manifold since the body parts are highly correlated [9]. We take advantage of the nice property and present *RefineBank (RB)* to reduce the noises in the motions. It learns an auto-encoder with a bank of latent codes to tightly represent the compact motion manifold. By projecting noisy or corrupted motions onto the manifold, we can remove the noises in the motions, which prevents error accumulation in the auto-regressive generation. Secondly, we propose *TransitBank (TB)* on top of the learned manifold, which maintains a past-future motion dynamics bank to provide explicit priors about the future motions. The priors narrow down the motion prediction space which facilitates high-fidelity motion generation with large magnitude. Mathematically, the prediction process can be formulated as:

$$\hat{\boldsymbol{d}}_{t+1:t+K} = \mathrm{RB}(\, P_c(\, E_m(\boldsymbol{m}_{1:t+K}),\, E_d(\hat{\boldsymbol{d}}_{1:t}),\, \mathrm{TB}(\hat{\boldsymbol{d}}_{1:t})\,)\,). \tag{2}$$

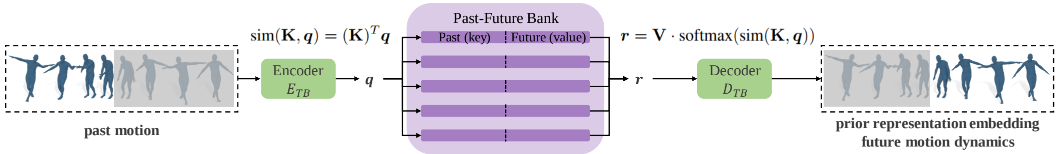


Figure 3: **TransitBank**: Given first half of the motion sequence, it produces a prior representation for the second half of the sequence by computing the weighted average of the values of the bank items to reduce the uncertainty and ambiguity in motion prediction.

4.1 RefineBank

Figure 2 shows the components of RefineBank which has an encoder E_{RB}, a decoder D_{RB}, and a manifold bank $\mathbf{B}_M \in \mathbb{R}^{C \times N}$. The bank $\mathbf{B}_M$ has N learnable latent codes of dimension C which span the low-dimensional dance manifold. As will be described later, the bank is learned from the GT motions so it can be interpreted as a prior probabilistic distribution where the motion data with noises will have small likelihood.

For a (noisy) motion sequence $\hat{\boldsymbol{d}}_{1:l}$ produced by the predictor, we first transform each of them to a latent feature by a transformer-based encoder E_{RB}. We uniformly sample n features at different time steps $\boldsymbol{z} = \left\{ \boldsymbol{z}_t | t \in \left\{ 0, \frac{l}{n-1}, \frac{2l}{n-1}, ..., l \right\} \right\}$ to represent the sequence. Then we project each $\boldsymbol{z}_t$ to the manifold represented by the bank to remove the prediction error. Specifically, for each latent feature $\boldsymbol{z}_t$, we compute the similarity scores between $\boldsymbol{z}_t$ and the latent codes in $\mathbf{B}_M$, and use the weights to project $\boldsymbol{z}_t$ onto the manifold to get $\hat{\boldsymbol{z}}_t$:

$$\hat{\boldsymbol{z}}_t = \mathbf{B}_M \cdot \mathrm{softmax}(\mathbf{B}_M^T \boldsymbol{z}_t). \tag{3}$$

The projected latent features $\{\hat{\boldsymbol{z}}_t\}$ will be fed to the transformer-based decoder D_{RB} to generate the motions that we expect to have little noise.

4.2 TransitBank

The high spatio-temporal complexity of the motion space increases the uncertainty and ambiguity of motion prediction. Hence, the motion predictors are prone to generating small-magnitude motions. To address the problem, we present TransitBank which is inspired by the fact that there exist a number of basic short motion segments that are frequently used in many dances. TransitBank aims to exploit such cues to estimate the prior distributions for future motion prediction.

Concretely, TransitBank works by dividing a motion sequence into past and future sub-sequences and storing them as past-future pairs. Then given the past motions as input, it queries the bank and reads the corresponding future motion dynamics from the bank. This prior information is fed to the cross-modal predictor to reduce the uncertainty and ambiguity in motion prediction.

Figure 3 shows the structure of TransitBank. It consists of an query encoder E_{TB}, a past-future bank B_{PF}, and a read decoder D_{TB}. The bank $\mathbf{B}_{PF} = \left\{ (\mathbf{K}, \mathbf{V}) | \mathbf{K}, \mathbf{V} \in \mathbb{R}^{C \times N} \right\}$ is also learned from training data. As for E_{TB} and D_{TB}, we use a simple transformer structure for encoder and decoder, respectively. To avoid information leakage in case that the cross-modal predictor directly copies the motions in TransitBank, we design the query-read process by using the attention mechanism rather than finding closest one which will effectively blur the future motions. Specifically, we encode input motion $\boldsymbol{d}_{1:l}$ via E_{TB} and take the average pooling of all outputs as the query vector $\boldsymbol{q}$. Then we compute the similarity between $\boldsymbol{q}$ and key features $\mathbf{K}$ as:

$$\mathrm{sim}(\mathbf{K}, \boldsymbol{q}) = (\mathbf{K})^T \boldsymbol{q}. \tag{4}$$

In the read process, we compute the future motion prior $\boldsymbol{r}$ as the weighted average of the corresponding values $\mathbf{V}$:

$$\boldsymbol{r} = \mathbf{V} \cdot \mathrm{softmax}(\mathrm{sim}(\mathbf{K}, \boldsymbol{q})). \tag{5}$$

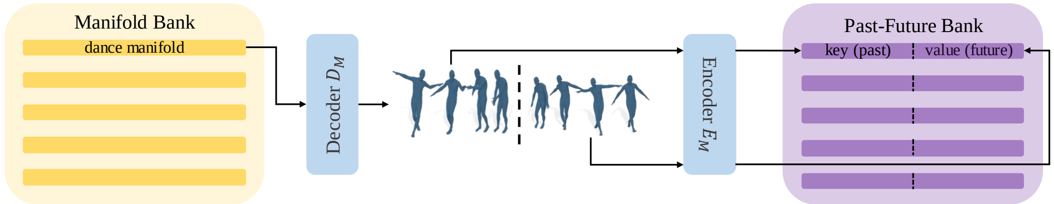


Figure 4: Constructing the past-future bank from the manifold bank.

Finally, we decode the vector $\boldsymbol{r}$ via D_{TB} and feed it to the cross-modal predictor as additional tokens.

4.3 Model Learning

This section describes how we learn the two banks from training data. It is worth noting that the two banks are not independent. Instead, TransitBank is a temporal extension of RefineBank. In the following part, we first illustrate how we learn the bank in RefineBank and then describe how to derive the one in TransitBank. Finally, we describe the training optimization strategy.

Bank in RefineBank We first initialize the bank by clustering. Specifically, we obtain a number of motion segments by applying a sliding window with length l to all training motion sequences. Then AP clustering [5] is used to obtain a number of cluster centers from the segments. We apply a transformer-based encoder E_M to encode each cluster center $\boldsymbol{d}^i_{1:l}$ and compute the average of the encoding outputs of the l frames to initialize one bank elements $\boldsymbol{b}^i \in \mathbf{B}_M$.

After initialization, we train the encoder E_M, decoder D_M and the bank $\mathbf{B}_M$ to reconstruct all motion segments $\boldsymbol{d}_{1:l}$ using the method described in Section 4.1. The only difference is that we project the latent feature $\boldsymbol{z}$ to its closest item $\boldsymbol{b}$ of $\mathbf{B}_M$ as:

$$\hat{\boldsymbol{z}} = \arg\min_{\boldsymbol{b}\in\mathbf{B}_M} \|\boldsymbol{z} - \boldsymbol{b}\| . \tag{6}$$

Bank in TransitBank Constructing $\mathbf{B}_{PF}$ from $\mathbf{B}_M$ is straightforward. As shown in Figure 4, for each element $\boldsymbol{b} \in \mathbf{B}_M$, we decode it to a motion sequence $\hat{\boldsymbol{d}}_{1:l}$ via the learned transformer decoder D_M. Then we divide the sequence into two parts: $\hat{\boldsymbol{d}}_{1:l/2}$ and $\hat{\boldsymbol{d}}_{l/2+1:l}$, which are fed to the encoder E_M to obtain the latent codes representing the past ($\mathbf{K}$) and future ($\mathbf{V}$), respectively. In that way, we obtain $\mathbf{B}_{PF} = \left\{(\mathbf{K}, \mathbf{V}) | \mathbf{K}, \mathbf{V} \in \mathbb{R}^{C\times N}\right\}$.

Optimization Strategy We train our model in three stages. In the first stage, we optimize the manifold bank by minimizing the following loss:

$$\mathcal{L}_{\text{ManifoldBank}} = \left\|\hat{\boldsymbol{d}}_{1:l} - \boldsymbol{d}_{1:l}\right\|_2^2 + \|\text{sg}[\boldsymbol{z}] - \hat{\boldsymbol{z}}\|_2^2 + \beta\|\boldsymbol{z} - \text{sg}[\hat{\boldsymbol{z}}]\|_2^2. \tag{7}$$

The first term minimizes the reconstruction error. The second part is the “item loss” [29] to update items in manifold bank, where sg denotes “stop gradient”. This objective function moves the items close to the outputs of the encoder. The sg[$\cdot$] operator is implemented by the identification function during forward computation with zero partial derivatives. The third part is “commitment loss” [29] to ensure the output of encoder commits to an item and the value of β is set as 0.2, empirically. We optimize the encoder with the first and third loss terms. The bank items are updated with only the second loss term and the decoder is trained with only the first term. Once we finalize the manifold bank, we can compute the past-future bank. Note that we do not update these two banks in the following stages.

In the second stage, we train the encoder and decoders in RefineBank to reconstruct all motion sequences of the training set with only reconstruction loss.

$$\mathcal{L}_{\text{RefineBank}} = \left\|\hat{\boldsymbol{d}}_{1:l} - \boldsymbol{d}_{1:l}\right\|_2^2, \tag{8}$$

In the third stage, we train the whole framework in an end-to-end manner with the L2 loss between predicted motion and ground truth motion.

$$\mathcal{L}_{\text{Prediction}} = \left\| \hat{\boldsymbol{d}}_{t+1:t+K} - \boldsymbol{d}_{t+1:t+K} \right\|_2^2, \tag{9}$$

5 Experiments

5.1 Experimental Settings

Dataset We evaluate our method on the largest AIST++ [18] dance dataset. It has 60 music pieces belonging to 10 dance genres. In total, there are 992 3D pose sequences at 60 FPS. Following [18], we use 952 samples for training and the rest 40 for evaluation.

Implementation Details The model takes 240 frames of music and 120 frames of motions as input and predicts the next $K = 20$ frames of motions. We use the same network structures for the music extractor, motion extractor and cross-modal predictor as FACT [18]. For the encoders and decoders in RefineBank and TransitBank, we use transformers with 4 layers and 10 attention heads with 2048 hidden size. The number of items in manifold bank and past-future bank is 256 and each item is a 2048-dim latent vector. The input to RefineBank and TransitBank is a motion sequence with $l = 120$ frames. Since $K < l$, we concatenate the previous $l - K$ frames of motions with the predictor output to feed to RefineBank. We construct a separate manifold bank and past-future bank for each dance genre. In the first training stage, we adopt Adam optimizer [13] with a learning rate of 1×10^{-4} to train the manifold bank for 50 epochs. In the second training stage, we pre-train the RefineBank using Adam optimizer with a learning rate of 1×10^{-4} for 25 epochs. In the third stage, we train the whole framework with Adam optimizer for 150 epochs. The learning rate starts with 1×10^{-4} and decreases to $\left\{1 \times 10^{-5}, 1 \times 10^{-6}\right\}$ after $\{30, 90\}$ epochs, respectively. The whole training process takes about four days on four 2080Ti GPUs.

Metrics The previous works mainly evaluate the dance generation results from three aspects: quality, diversity and alignment. Following [18], we compute FID [7] (Frechet Inception Distances) on the kinetic features (denoted as $\mathbf{FID}_k$) and geometric features (denoted as $\mathbf{FID}_g$), respectively, to measure quality. We use the fairmotion toolbox [6] to extract the features. For diversity, we compute the average Euclidean distance in the kinetic (denoted as $\mathbf{Dist}_k$) and geometric (denoted as $\mathbf{Dist}_g$) feature space across the generated motions. For dance-music alignment, we adopt **Beat Alignment Score** in [18] to compute average distance between every kinematic beat and its nearest music beat. Since the freezing problem is largely overlooked previously, there are no metrics available to evaluate. We propose to compute the average values of the temporal differences of the pose and translation parameters in the whole sequence, which are termed as Δ_{Pose} and Δ_{Trans}, respectively. In addition, we also calculate the **Freezing Rate** of each sequence. We divide a sequence into non-overlapping sub-sequences of 60 frames, and for each sub-sequence, if $\Delta_{\text{Pose}} \leq \Delta^{\text{gt}}_{\text{Pose}}$ and $\Delta_{\text{Trans}} \leq \Delta^{\text{gt}}_{\text{Trans}}$ where Δ^{gt} is a predefined threshold statistically derived from the training set, we regard it as a freezing sub-sequence. Then we compute the percentage of freezing sub-sequences.

5.2 Comparison to the State-of-the-arts

We compare our approach to a number of recent methods including Li et al. [17], Dancenet [38], DanceRevolution [10], and FACT [18]. Our approach employs the same structure as FACT except that it has RefineBank and TransitBank. For each music, we generate a motion sequence with 1200 frames (20 seconds). The experiment results are shown in Table 1. Since DanceRevolution [10] predicts 3D keypoint positions, we cannot compute the SMPL-based freezing metrics. The approach in [17] does not predict the translation parameters. As shown, our approach outperforms the state-of-the-art methods on all metrics. Interestingly, the Δ_{Trans} of our method is even larger than that of GT. We visually compare our generated motions and GT motions and find that it is because GT often have stationary poses at transition moments. By contrast, it is difficult for learning-based methods to predict stationary poses due to the lack of sufficient data.

We present a detailed analysis for one sequence in Figure 5. As we can see, the motion and translation changes of FACT gradually decrease to a small number suggesting that the freezing situation

Table 1: Comparison to the state-of-the-art methods on the AIST++ dataset.

	Quality					Diversity		Align	User Study
Method	$\text{FID}_k \downarrow$	$\text{FID}_g \downarrow$	$\Delta_{\text{Pose}} \uparrow$	$\Delta_{\text{Trans}} \uparrow$	Freeze $\downarrow$	$\text{Dist}_k \uparrow$	$\text{Dist}_g \uparrow$	BeatAlign $\uparrow$	Win Rate $\uparrow$
GT	-	-	3.28	1.16	18.7%	9.06	7.31	0.292	31.7%
Li et al. [17]	86.43	20.58	1.02	-	59.0%	6.85	4.93	0.232	95.8%
DanceNet [38]	69.18	17.76	1.25	0.80	46.8%	2.86	2.72	0.232	90.8%
Revolution [10]	73.42	31.01	-	-	-	3.52	2.46	0.220	84.2%
FACT [18]	35.35	22.11	1.33	1.07	39.0%	5.94	6.18	0.241	86.7%
Ours	**25.96**	**14.42**	**1.64**	**1.36**	**29.6%**	**7.68**	**6.59**	**0.249**	-

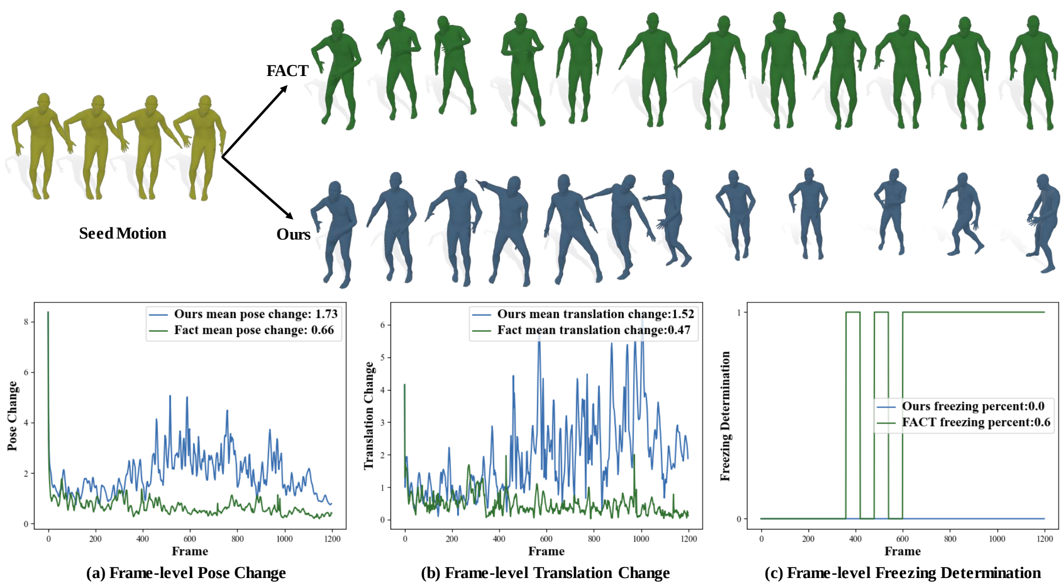


Figure 5: The top row shows the motions generated by FACT and our method for the same music. The bottom provides frame-level statistics for the two sequences. In figure (c), 0 represents non-freezing and 1 means freezing. Best viewed in color.

occurs. In contrast, the numbers of our method are always above those of FACT and do not freeze. The results validate that our proposed bank-based manifold projection and past-future dynamic priors indeed improve the quality of the generated motions.

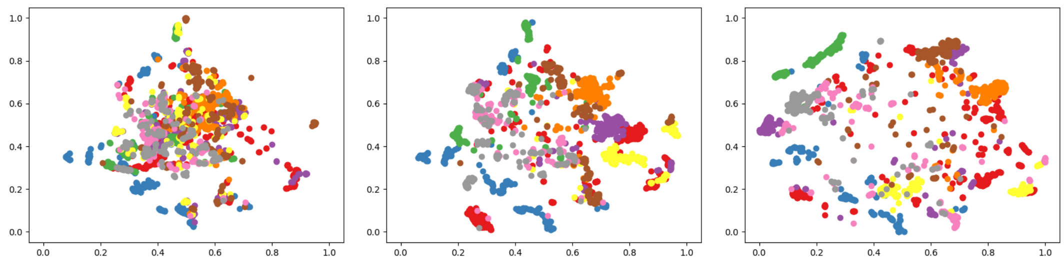
User Study We conduct a user study to investigate how people think of the dances generated by our method and the other ones. We invite 20 participants and each participant is asked to watch 30 pairs of comparison videos. Each pair consists of our dance and one competitor's generated with the same music. We ask each participant to determine "which person is dancing better to the music" and provide the statistics in Table 1 (last column). Our approach significantly outperforms the other methods in user study. We can keep at least 84.2% win rate to the SOTA methods including DanceRevolution and FACT. Notably, we can achieve 31.7% win rate compared to GT motions. We also collect detailed feed-backs and find that our generated dance is generally thought to be more diverse and non-freezing. The main problem with FACT is that the motions freeze frequently while the problem with DanceRevolution is that the motions are unnatural. Compared to GT motions, ours are thought to lack suitable transition motions and precise beat alignment which is a general problem faced by most prediction-based methods.

5.3 Ablation Study

Ablation Study of RefineBank and TransitBank The experiment results are shown in Table 2. Our first observation is that adding RefineBank to the baseline notably improves Δ_{Pose} meaning

Table 2: Ablation study of RefineBank and TransitBank.

Method	Quality					Diversity		Alignment
	$\text{FID}_k \downarrow$	$\text{FID}_g \downarrow$	$\Delta_{\text{Pose}} \uparrow$	$\Delta_{\text{Trans}} \uparrow$	Freezing $\downarrow$	$\text{Dist}_k \uparrow$	$\text{Dist}_g \uparrow$	BeatAlign $\uparrow$
Baseline	35.35	22.11	1.33	1.07	39.0%	5.94	6.18	0.241
+ RefineBank	28.67	16.38	1.53	1.15	32.1%	6.65	6.34	0.246
+ TransitBank	31.24	19.18	1.49	1.31	34.3%	7.42	6.47	0.245
Ours	**25.96**	**13.42**	**1.64**	**1.36**	**29.6%**	**7.68**	**6.59**	**0.249**



(a) Baseline (b) With Only RefineBank (c) With RefineBank & TransitBank

Figure 6: t-SNE visualization of the generated dances. Each dot represents a 3D pose and different colors represent different genres of the music used to generate the poses. Best viewed in color.

that the method reduces the chance of getting freezing motions. Meanwhile, the improvement also brings benefits to dance quality and diversity. However, the variation of the global positions of the dancers, *i.e.*, Δ_{Trans}, is only slightly improved. This is expected since RefineBank only guarantees that the refined motions are on the manifold. In contrast, adding TransitBank notably improves Δ_{Trans}. Meanwhile the diversity metrics are also notably improved. The results suggest that by exploiting the motion dynamic priors the method can predict high-fidelity and diverse motions with large magnitude instead of mean poses. Finally, the two modules are complementary to each other and combining them will further improve the results on all metrics.

We visualize the poses in the generated dances using t-SNE [30] in Figure 6. We can see that the dances generated by the baseline tend to be mixed together with other genres. It means the motions may have lower fidelity, smaller magnitude and lack uniqueness. We think this is caused by the freezing problem and the high spatio-temporal complexity of the prediction space. In contrast, adding RefineBank alleviates the freezing problem which allows the poses to preserve the high-fidelity details and to be differentiable from other dances. Further adding TransitBank allows to generate more diverse dances with larger motion magnitude.

Bank-based Auto-encoder We compare our bank-based auto-encoder with other options such as vanilla AE and VAE. The experimental results are shown in Table 3. Our bank-based auto-encoder achieves clearly better results than the other methods. This is because the bank of latent codes provide a tight characterization of the dance manifold. The tightness requires that the generated motion sequences strictly follow the dance styles.

We also compare to a discrete variant of Bank-AE. Different from our current method, it uses the closest bank item instead of convex combinations of the neighboring items to reconstruct each datum similar to VQ-VAE [29]. We can see that it also reduces the the freezing rate. However, the quality and diversity metrics are notably worse than our method. This is because the discrete variant has limited capability to reconstruct data with sufficient accuracy, compared to our approach using convex combinations of multiple bank items.

Number of Latent Features We study the impact of the number of latent features n for representing a motion segment as discussed in section 4.1. The results are shown in Table 4. In general, using more latent features improves the prediction performance because it can capture more details. But keep increasing the number may lead to degeneration.

Table 3: Comparison of the bank-based AE and other options.

Method	Quality					Diversity		Align
	$\text{FID}_k \downarrow$	$\text{FID}_g \downarrow$	$\Delta_{\text{Pose}} \uparrow$	$\Delta_{\text{Trans}} \uparrow$	Freezing $\downarrow$	$\text{Dist}_k \uparrow$	$\text{Dist}_g \uparrow$	BeatAlign $\uparrow$
AE	29.85	17.64	1.52	1.32	33.5%	7.48	6.49	0.245
VAE	29.67	17.05	1.53	1.32	33.1%	7.49	6.50	0.245
Bank-AE (Discrete)	27.48	15.19	1.60	1.34	30.4%	7.45	6.48	0.247
Bank-AE (Ours)	**25.96**	**13.42**	**1.64**	**1.36**	**29.6%**	**7.68**	**6.59**	**0.249**

Table 4: Evaluation on the number of latent feature.

	Quality					Diversity		Alignment
	$\text{FID}_k \downarrow$	$\text{FID}_g \downarrow$	Pose $\uparrow$	Trans $\uparrow$	Freezing $\downarrow$	$\text{Dist}_k \uparrow$	$\text{Dist}_g \uparrow$	BeatAlign $\uparrow$
n=1	27.94	15.61	1.59	1.34	31.2%	7.51	6.52	0.246
n=2	26.72	14.78	1.62	1.35	30.6%	7.63	6.57	0.248
n=3	26.15	13.95	1.64	1.36	29.9%	7.66	6.59	0.249
n=4	**25.96**	**13.42**	**1.64**	**1.36**	**29.6%**	7.68	6.59	**0.249**
n=5	26.67	13.88	1.63	1.35	30.1%	**7.72**	**6.63**	0.248

Table 5: Evaluation on the number of bank items.

	Quality					Diversity		Alignment
	$\text{FID}_k \downarrow$	$\text{FID}_g \downarrow$	Pose $\uparrow$	Trans $\uparrow$	Freezing $\downarrow$	$\text{Dist}_k \uparrow$	$\text{Dist}_g \uparrow$	BeatAlign $\uparrow$
N=32	29.74	17.34	1.53	1.32	32.9%	7.50	6.49	0.245
N=64	29.14	16.98	1.54	1.33	31.7%	7.56	6.52	0.246
N=128	27.31	14.86	1.61	1.35	30.2%	7.62	6.54	0.248
N=256	**25.96**	**13.42**	**1.64**	**1.36**	**29.6%**	**7.68**	**6.59**	**0.249**
N=512	26.39	14.07	1.63	1.35	29.9%	7.64	6.56	0.248

Number of Bank Items We study the impact of the number of elements N in the two banks. The results are shown in Table 5. Initially, increasing the number of elements improves the results. This is reasonable because the expressive power of the bank is improved and more details can be preserved after projection. However, keep increasing N to 512 begins to have negative effects. We think this is because using too many items may increase the risk of over-fitting to the small levels of noises in the GT data. In addition, the introduced redundancy may also bring negative effects. Nevertheless, the method is relatively robust to this parameter and achieves reasonably good results when N is between 128 and 512.

6 Conclusion

In this work, we presented two general modules that can be plugged into the existing methods to address the freezing problem in dance motion generation. This largely overlooked problem has limited motion generation to short segments of only several seconds. By reducing noise accumulation and exploiting dynamic priors, our approach can generate motions for at least 30 seconds with 60 FPS, which is the maximum music length in current dataset AIST++, without freezing. Our user study also shows that our method has obvious advantages over other methods in terms of quality and diversity. The method paves the way for addressing a more valuable problem of full choreography for entire musics instead of short clips.

Broader Impact We believe that our work has values for not only dance generation but also for more general motion prediction. This benefits areas including media platforms, robotics, and autonomous driving. On the other hand, our method can have negative downstream consequences such as being extended to generate fake videos conditioned on the generated human motions with GANs. The potential limitation is that due to the relatively short duration of music pieces in AIST++ dataset, the test performance is not a precise evaluation for long-term dance generation capability.

References

[1] Omid Alemi, Jules Françoise, and Philippe Pasquier. Groovenet: Real-time music-driven dance movement generation using artificial neural networks. *networks*, 8(17):26, 2017.

[2] Kang Chen, Zhipeng Tan, Jin Lei, Song-Hai Zhang, Yuan-Chen Guo, Weidong Zhang, and Shi-Min Hu. Choreomaster: choreography-oriented music-driven dance synthesis. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021.

[3] Yinglin Duan, Tianyang Shi, Zhipeng Hu, Zhengxia Zou, Changjie Fan, Yi Yuan, and Xi Li. Automatic translation of music-to-dance for in-game characters.

[4] Joao P Ferreira, Thiago M Coutinho, Thiago L Gomes, José F Neto, Rafael Azevedo, Renato Martins, and Erickson R Nascimento. Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio. *Computers & Graphics*, 94:11–21, 2021.

[5] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[6] Deepak Gopinath and Jungdam Won. fairmotion - tools to load, process and visualize motion capture data. Github, 2020.

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[8] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016.

[9] Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 technical briefs*, pages 1–4. 2015.

[10] Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Daxin Jiang. Dance revolution: Long-term dance generation with music via curriculum learning. In *International Conference on Learning Representations*, 2020.

[11] Yuhang Huang, Junjie Zhang, Shuyan Liu, Qian Bao, Dan Zeng, Zhineng Chen, and Wu Liu. Genre-conditioned long-term 3d dance generation driven by music. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4858–4862. IEEE, 2022.

[12] Jae Woo Kim, Hesham Fouad, and James K Hahn. Making them dance. In *AAAI Fall Symposium: Aurally Informed Performance*, volume 2, 2006.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Kosmas Kritsis, Aggelos Gkiokas, Aggelos Pikrakis, and Vassilis Katsouros. Danceconv: Dance motion generation with convolutional networks. *IEEE Access*, 2022.

[15] Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. *Advances in Neural Information Processing Systems*, 32, 2019.

[16] Buyu Li, Yongchi Zhao, Zhelun Shi, and Lu Sheng. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. AAAI, 2022.

[17] Jiaman Li, Yihang Yin, Hang Chu, Yi Zhou, Tingwu Wang, Sanja Fidler, and Hao Li. Learning to generate diverse dance motions with transformer. *arXiv preprint arXiv:2008.08171*, 2020.

[18] Ruilong Li, Shan Yang, David A Ross, and Angjoo Kanazawa. Ai choreographer: Music conditioned 3d dance generation with aist++. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13401–13412, 2021.

[19] Wei Mao, Miaomiao Liu, and Mathieu Salzmann. Generating smooth pose sequences for diverse human motion prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13309–13318, 2021.

[20] Brian McFee, Colin Raffel, Dawen Liang, Daniel P Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25. Citeseer, 2015.

[21] Ferda Ofli, Yasemin Demir, Yücel Yemez, Engin Erzin, A Murat Tekalp, Koray Balcı, İdil Kızoğlu, Lale Akarun, Cristian Canton-Ferrer, Joëlle Tilmanne, et al. An audio-driven dancing avatar. *Journal on Multimodal User Interfaces*, 2(2):93–103, 2008.

[22] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021.

[23] Xuanchi Ren, Haoran Li, Zijian Huang, and Qifeng Chen. Self-supervised dance video synthesis conditioned on music. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 46–54, 2020.

[24] Takaaki Shiratori and Katsushi Ikeuchi. Synthesis of dance performance based on analyses of human motion and music. *Information and Media Technologies*, 3(4):834–847, 2008.

[25] Takaaki Shiratori, Atsushi Nakazawa, and Katsushi Ikeuchi. Dancing-to-music character animation. In *Computer Graphics Forum*, volume 25, pages 449–458. Wiley Online Library, 2006.

[26] Guofei Sun, Yongkang Wong, Zhiyong Cheng, Mohan S Kankanhalli, Weidong Geng, and Xiangdong Li. Deepdance: music-to-dance motion choreography with adversarial learning. *IEEE Transactions on Multimedia*, 23:497–509, 2020.

[27] Jiangxin Sun, Zihang Lin, Xintong Han, Jian-Fang Hu, Jia Xu, and Wei-Shi Zheng. Action-guided 3d human motion prediction. *Advances in Neural Information Processing Systems*, 34, 2021.

[28] Taoran Tang, Jia Jia, and Hanyang Mao. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 1598–1606, 2018.

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

[30] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[31] Nelson Yalta, Shinji Watanabe, Kazuhiro Nakadai, and Tetsuya Ogata. Weakly-supervised deep recurrent neural networks for basic dance step generation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[32] Sijie Yan, Zhizhong Li, Yuanjun Xiong, Huahan Yan, and Dahua Lin. Convolutional sequence generation for skeleton-based action synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4394–4402, 2019.

[33] Zijie Ye, Haozhe Wu, Jia Jia, Yaohua Bu, Wei Chen, Fanbo Meng, and Yanfeng Wang. Choreonet: Towards music to dance synthesis with choreographic action unit. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 744–752, 2020.

[34] Ye Yuan and Kris Kitani. Dlow: Diversifying latent flows for diverse human motion prediction. In *European Conference on Computer Vision*, pages 346–364. Springer, 2020.

[35] Jason Y Zhang, Panna Felsen, Angjoo Kanazawa, and Jitendra Malik. Predicting 3d human dynamics from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7114–7123, 2019.

[36] Xinjian Zhang, Yi Xu, Su Yang, Longwen Gao, and Huyang Sun. Dance generation with style embedding: Learning and transferring latent representations of dance styles. *arXiv preprint arXiv:2104.14802*, 2021.

[37] Yi Zhou, Zimo Li, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. In *International Conference on Learning Representations*, 2018.

[38] Wenlin Zhuang, Congyi Wang, Jinxiang Chai, Yangang Wang, Ming Shao, and Siyu Xia. Music2dance: Dancenet for music-driven dance generation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(2):1–21, 2022.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See section 6
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See section 6
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Code and models will be released upon acceptance to ensure the reproducibility.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See section 5.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Results reported in this paper are the mean results for three repeating experiments.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See section 5.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See section 5.
   (b) Did you mention the license of the assets? [Yes] We provide the license in the supplemental material.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] See section 5.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [No]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] Considering that user study can be completed in less than half an hour, we offer each participant a payment of 10 dollars.