Mining and Applying Composition Knowledge of Dance Moves for Style-Concentrated Dance Generation

Xinjian Zhang[1,2], **Su Yang**[1,2*], **Yi Xu**[1,2], **Weishan Zhang**[3], **Longwen Gao** [4]
[1]School of Computer Science, Fudan University
[2]Shanghai Key Laboratory of Intelligent Information Processing
[3]Department of Software Engineering, China University of Petroleum (East China)
[4]Bilibili
{zhangxj17, suyang, yxu17}@fudan.edu.cn, zhangws@upc.edu.cn, gaolongwen@bilibili.com

Abstract

Choreography refers to creation of dance motions according to both music and dance knowledge, where the created dances should be style-specific and consistent. However, most of the existing methods generate dances using the given music as the only reference, lacking the stylized dancing knowledge, namely, the flag motion patterns contained in different styles. Without the stylized prior knowledge, these approaches are not promising to generate controllable style or diverse moves for each dance style, nor new dances complying with stylized knowledge. To address this issue, we propose a novel music-to-dance generation framework guided by style embedding, considering both input music and stylized dancing knowledge. These style embeddings are learnt representations of style-consistent kinematic abstraction of reference dance videos, which can act as controllable factors to impose style constraints on dance generation in a latent manner. Hence, we can make the style embedding fit into any given style while allowing the flexibility to generate new compatible dance moves by modifying the style embedding according to the learnt representations of a certain style. We are the first to achieve knowledge-driven style control in dance generation tasks. To support this study, we build a large multi-style music-to-dance dataset referred to as I-Dance. The qualitative and quantitative evaluations demonstrate the advantage of the proposed framework, as well as the ability to synthesize diverse moves under a dance style directed by style embedding.

1 Introduction

Dancers move elegantly following the rhythm of music. Behind dancing is the hard labor in terms of choreography. Consequently, automatic music-to-dance choreography has emerged as a new topic in multimedia (Lee et al. 2019; Ye et al. 2020; Ren et al. 2020; Siyao et al. 2022). It can enable various real applications, such as virtual character generation, game creation, and teaching assistant. Specifically, investigations on music-to-dance synthesis give rise to a new problem, say, machine perception of the latent knowledge regarding dance moves as well as matching rhythm between music and dance in terms of beat and intensity, that is, style.

Historically, most dance generation methods pay efforts to model the relationship between music and dance in feature space or latent space without any stylized knowledge. Early studies (Shiratori, Nakazawa, and Ikeuchi 2006; Ofli et al. 2008; Fan, Xu, and Geng 2011; Asahina et al. 2016) are mainly retrieval based, which are focused on selecting the best matching pair from the database based on music and dance features. Due to the retrieval nature, these methods cannot generate new dance motions. In recent years, deep generative models (Alemi, Françoise, and Pasquier 2017; Tang, Jia, and Mao 2018; Lee et al. 2019; Ye et al. 2020; Ren et al. 2020; Huang et al. 2021; Siyao et al. 2022) are introduced to alleviate this problem. The temporal indexes (Tang, Jia, and Mao 2018) and contrastive cost function (Ren et al. 2020) have been utilized to promote the model's ability of generation. However, such methods usually regress to deviating from normal dance moves, for example, remaining rigid or shaking meaninglessly. To deal with this problem, (Lee et al. 2019; Ye et al. 2020) attempt to label dances into a series of basic dance units manually and learn to compose a dance by organizing multiple dancing units, which cost tremendous manual efforts and are unable to be compatible with different rhythms. Yet, machine-generated choreography is far behind human works in terms of coherence to a desired style because the professional knowledge behind choreography, such as dance styles as well as the the motion patterns favored by each given dance style, is not exploited in the previous works. In view of that, this study aims to discover from a large corpus of dance videos the basic motion patterns and how a certain dance style is subject to such patterns so that style-controllable dance generation can be

Figure 1: The example of dances in different styles accompanying with the same music. All dancers' moving rhythm match the music well.

*Corresponding author.

approached by applying such composition knowledge of a given dance style.

In fact, human choreographers with different dance training experiences can compose dances of various styles. As shown in Figure 1, the dance moves accompanying the same music can be composed of sharp tuning with diverse style variations. The previous methods struggle to establish a one-to-one mapping for music and dance without any stylized knowledge, which limits the model's ability to generate diversely new dance moves coherent to a desired style. Provided a model can learn meaningful stylized representations from known dance moves so as to direct the dance generation process, the generated dance should be more realistic and diverse to make a style hold vividly and creatively. So far, in the context of automatic dance generation, how to learn representations of dance styles as well as the composition of basic dance moves has been remaining a missing topic, which is the main target to be tackled in this study.

In order to learn aforementioned stylized dancing knowledge contained in dance moves, we introduce a self-attention vector quantized (VQ) (Oord, Vinyals, and Kavukcuoglu 2017) encoder in our dance generation framework. It learns dance styles in a latent space spanned by a base of prototypes, which form a codebook of basic motion patterns shared by all the dance styles of interest. The stylized knowledge is computed automatically as a linear combination of such prototype vectors with variable weights to indicate different styles, referred to as style embedding, where the weights corresponding with a specific dance style can be taught by realistic dance videos of this style, and alerting configuration of such weights means changing to favor different basic dance moves, leading to style-governed dance move creation as diversely as one desires. At the dance generation stage, we design a set of transformer encoders (Vaswani et al. 2017) to obtain the temporal representation of input music. Then, a long short-term memory (LSTM) module incorporates both the music representation and the style embedding to generate dance in an autoregressive way. Benefiting from the learnable and transferable style embedding to indicate the dance style favoring certain prototypes, our framework is capable of creating diverse dance moves of any desired style by modifying the style embedding to explore the best extent of this style.

To support this study, we construct a large music-to-dance dataset with style annotation for training and evaluation, referred to as I-Dance, where three modern dancing categories: Anime dance, popping dance, and locking dance, are selected to build the dataset since they have a higher degree of freedom in terms of choreography.

Our contributions are summarized as follows: (1) We propose a novel dance generation framework using style embedding to guarantee that the generated results are coherent to the coach in terms of style, which allows flexible style manipulation. (2) We propose a representation learning scheme to model a dance style as a set of weights to render a linear combination of some basic dance moves, namely, prototypes, which are learnt from real dances in an end-to-end manner. Then, the latent knowledge in the form of the style-related weighting of the prototypes is transferred to dance generation so as to guarantee style consistency between the real and generated dances while allowing flexible modification of the style representation to create new dance moves of diversity. (3) We establish a large music-to-dance dataset that contains choreographies of three dance styles. (4) The qualitative and quantitative results validate that the proposed framework can generate more diverse and realistic dance moves under a given style, adapting to the same input music.

2 Related Work

2.1 Music-to-dance Generation

The music-to-dance generation models can be roughly divided into two categories: Retrieval-based methods and learning-based methods. The previous works (Shiratori, Nakazawa, and Ikeuchi 2006; Kim et al. 2009; Fan, Xu, and Geng 2011; Lee, Lee, and Park 2013) on dance generation carefully design musical features to retrieve the dance moves from a database tied to the musical features. These retrieval-based methods suffer from lack of machine learning to capture the generic correlation between music and dance and cannot generate novel dances. Recently, (Alemi, Françoise, and Pasquier 2017; Tang, Jia, and Mao 2018; Lee et al. 2019; Ye et al. 2020; Ren et al. 2020; Sun et al. 2020; Li et al. 2021; Huang et al. 2021; Ferreira et al. 2021) attempt to apply deep generative models to generate more creative dances. (Yalta et al. 2019; Tang, Jia, and Mao 2018) design two LSTM Auto-Encoder models. However, these methods are not suitable for long-course dance generation due to the local attention nature, so the transformer-based methods (Huang et al. 2021; Li et al. 2021; Siyao et al. 2022) appear. Even though, a part of these methods will regress to nonstandard poses deviating from any known dance style, due to lack of explicitly spatial constraint between body parts that complies with a style. Hereafter, (Ye et al. 2020; Lee et al. 2019) decompose a dance into dance units conditional on music beats and let the models learn to recompose them according to input music. Being less aware of the knowledge regarding style-related dance moves, such methods require training different model parameters with regard to different dance styles. Solutions as such are not efficient, nor generic.

As all these methods fail to do dance generation under the control of prior knowledge regarding style-related dance move composition, such a missing topic motivates this study to render dance generation from knowledge discovery perspective, that is, how to deploy learnable representation to figure out style-favored dance moves and apply the mined knowledge to direct style-controllable dance generation.

2.2 Vector Quantized Variational Autoencoder

The Vector Quantized Variational Autoencoder (VQ-VAE) (Oord, Vinyals, and Kavukcuoglu 2017) adds a vector quantization (VQ) layer in the variational autoencoder (VAE) (Kingma and Welling 2013) to produce a discrete latent representation. Pairing it with an autoregressive prior, the model can generate high-quality images (Razavi, van den Oord, and Vinyals 2019; Peng et al. 2021), videos (Yan et al. 2021), and speech (Gârbacea et al.

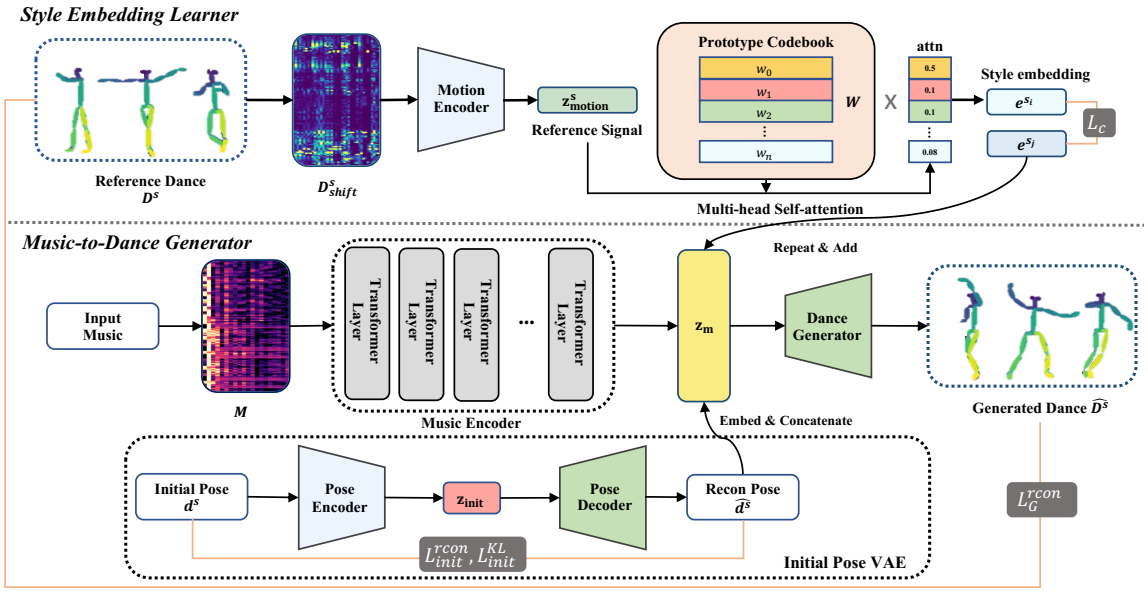

Figure 2: The architecture of the proposed dance generation framework with prototype codebook supported style embedding.

2019; Wang et al. 2018b), as well as high-quality speaker conversion with unsupervised learning of phonemes. These provide further evidence that the VQ-VAE can efficiently capture the high-level features, yielding a meaningful representation of the input. In this paper, inspired by VQ-VAE, we design an improved VQ encoder with self-attention to fit the style embedding of the reference dances in terms of dance style, which is the first attempt to apply modified VQ-VAE in this context.

3 Dataset and Preprocessing

The widely used music-to-dance generation dataset is AIST++ (Li et al. 2021), which contains about $0.23M$ advanced dance frames (calculated in 30fps) of non-repeating dances. It can not satisfy current complex models well. Therefore, we build a large multi-style music-to-dance dataset with three popular representative categories: Popping dance, anime dance, and locking dance. All dance clips in the dataset are performed by professional dancers, with clear high-quality music. The Dance clips are resized to 640×480. Our novel dataset contains 70 popping dance videos, 150 anime dance videos, and 65 locking dance videos, with $1.48M$ selected frames (30fps), much larger than AIST++. No personally identifiable information will include in the I-Dance dataset.

Musical feature extraction. We first normalize the music volume according to the root mean square. Then, the musical features are extracted via an audio analysis library Librosa (McFee et al. 2015), including 20-dimensional Mel Frequency Cepstrum Coefficients (MFCC), 12-dimensional constant-Q transform chromagram, 1-dimensional pitch, 1-dimensional root-mean-square energy, 1-dimensional beat, and 1-dimensional onset strength.

Motion feature acquisition. All raw videos are fed into Openpose (Cao et al. 2017) for 2D keypoints detection. We manually filter out the video clips whose skeletons are not detected correctly to assure that the skeleton information in each dance clip is of high quality. We finally choose 21 joints most relevant to kinematic expressiveness to represent the dance, including the nose, neck, mid-hip, left and right ears, shoulders, elbows, wrists, hips, knees, hand, and ankles.

4 Methodology

Aside from realistic-effect dance generation, another important but missing issue is style-controllable dance generation, which means not only fitting well into the video teacher in terms of dance style but also applicable to generate diverse moves under such style control. To realize this goal, it contains 4 key steps: (1) Develop a learnable framework to represent kinematic body motion in a highly abstractive manner, namely, style embedding, which can act as a control signal to direct dance generation. (2) Let such learnable representation figure out a dance style that favors specific dances moves by fitting it into given coach videos through machine learning. (3) Reuse the representation that have been taught to be subject to the teaching videos to generate style-consistent dances. (4) Alter the style embedding to obtain new dance moves diversely while keeping the style embedding an inlier of the category that exerts style control.

As illustrated in Figure 2, we propose a novel end-to-end dance generation framework with a learnable representation referred to as style embedding to learn the prototypes corresponding to basic dance moves as well as how much such prototypes are favored in the coach video of a certain style, where the weights in association with the prototype vectors in the linear combination of them figure out the so-called style. As the generated dance should correspond to music in terms of rhythm, and move continuously from the original pose, a music-to-dance generation model to map musical features to dance motions is deployed to work in collabora-

tion with the style embedding. In knowledge mining phase, the style embedding learner acts to summarize the attributes of the reference dance motions into a vector indicating the dance style of interest. In style-transferring phase, the latent knowledge in the form of style embedding with varying configuration of the weights of prototypes can alter the style from slightly to remarkably, visiting the extent of it. The dance generator finally generates dances conditioned on the music representation, the style embedding, and the initial pose resulting from a VAE model (Kingma and Welling 2013). For a task of specific-style dance generation, we can easily apply the style embedding learnt from the coach video to the music-to-dance generation model. Moreover, we can alter the values of the style embedding to control the style so as to enable diversity of dance moves, and under the control of style embedding, the music-to-dance generation model can generate dances of any expected style.

In the following, we first present the mechanism of style embedding. Then, we demonstrate a contrastive loss that forces the learnt representations of style embeddings to fall into discriminative clusters representing different styles. Finally, the overall music-to-dance generator subject to style embedding is described.

4.1 Prototype-equipped Dance Style Learner

In fact, human's dance style knowledge is established in long-term practice of perception of dance motions. Such memory of dance styles is in general the composition of prototype patterns. Motivated by this, we represent the style embedding as a linear combination of a couple of prototype patterns. The prototype patterns are learnt from the whole corpus of real dance, and one solution of the weights in association with such prototype patterns in terms of a linear combination of them indicates a specific dance style, which are solved to fit well into the video examples of a certain style. Here, the configuration of the weights flags a dance style in that how much each prototype is favored in this style. This process is conceptually similar to the encoder in VQ-VAE (Oord, Vinyals, and Kavukcuoglu 2017). Unlike the general implementation of VQ-VAE, we replace the nearest measure method with multi-head self-attention. In the following, we present how to learn the prototype patterns and represent style embedding through the weighted sum of such prototypes in an end-to-end manner.

As shown in the style embedding learner in Figure 2, the reference dance is formulated as $\{D_t^s \in \mathbb{R}^{2J} | t = 1, \ldots, T\}$ with a style indicator s, where $J = 21$ indicates all joint locations of interest. The style embedding learner first computes the shift between two sequential motion frames as dynamic motion feature: $D_{shift}^s = \{D_{t+1}^s - D_t^s\}, t = 1, \ldots, T$, where t indicates the time. In the motion encoder, a stack of 2-D convolutional layers and a GRU layer map the dynamic motion feature D_{shift}^s into a fixed-length motion representation $z_{motion}^s \in \mathbb{R}^{1 \times d_{gru}}$, which is the last state of the GRU layer, where d_{gru} is the hidden size of the GRU layer. Hereafter, z_{motion}^s encodes the reference dance motion's spatial and temporal features in a dance clip, serving as the reference to induce the style knowledge.

Then, we design a learnable matrix referred to as the codebook to learn the prototype patterns from the motion representations of the whole dance corpus including all dance styles of interest, so as to discover though machine learning the basic dance patterns shared by all styles, namely, prototypes. As shown in Figure 2, we feed D_{shift}^s to the style embedding learner to extract the motion representation z_{motion}^s, and based on such representation, we learn a codebook $W \in \mathbb{R}^{d_w \times d_{gru}}$ to memorize all prototypes in the training phase, where d_w is a hyper-parameter to determine the size of the codebook, that is, the number of prototypes. Here, each row of W can be regarded as a prototype vector, which is a motion descriptor in the form of z_{motion}^s to span a latent space. By means of W, any given motion pattern can be represented as a linear combination of the d_w prototype vectors stored in W with the weights of such linear combination learnable, where a solution of these weights indicates how frequently each prototype motion pattern is favored in a given dance style. In learning the representation of the dance style from a coach video, we leverage the multi-head self-attention scheme to calculate the similarity between a given motion pattern z_{motion}^s in the coach video and each prototype vector in W to demonstrate how much it is close to each prototype vector. The multi-head self-attention can calculate the similarity in different subspace via the fully connected layers at different head, and we set the number of head H to 4 in practice. The weighted sum of the prototype vectors, namely style embedding, is passed to the music-to-dance generator module as a control signal at every time step. Here, the style embedding $e^s \in \mathbb{R}^{1 \times d_{gru}}$ is calculated as follow:

$$attn_h = \frac{Softmax[FC_z^h(z_{motion}^s) FC_w^h(W^T)]}{\sqrt{d_{gru}}},$$
$$e^s = Mean\{\sum_{h=1}^{H} [attn_h FC_w^h(W)]\}, \quad (1)$$

where FC_z^h and FC_w^h are the fully connected layer imposed on z_{motion}^s and W, respectively, and h indicates the head. $attn_h \in \mathbb{R}^{1 \times d_w}$ calculated by the attention scheme functions to weight the contribution of each prototype pattern. Note that both the prototype vectors representing the basic motion patterns and the weights to combine them into the style embedding through linear combination are learnt end-to-end from coach videos in framework shown in Figure 2.

4.2 Learning Representations Converging to Style-related Clusters via a Contrastive Loss

The loss function plays an important role in learning the prototype patterns and their weights to form the style embedding. A proper loss function makes the learnt representations of dance styles fall into compact and separable clusters, while a bad one will cause overlap of the representations between different style categories.

To enable visualization of the high-dimensional style embeddings $\{e^s\}$ of all the samples in the I-Dance dataset, we reduce the style embeddings' dimension to 3 via principal components analysis (PCA) in order to observe the relationship between them. As shown in Figure 3 (a), without any

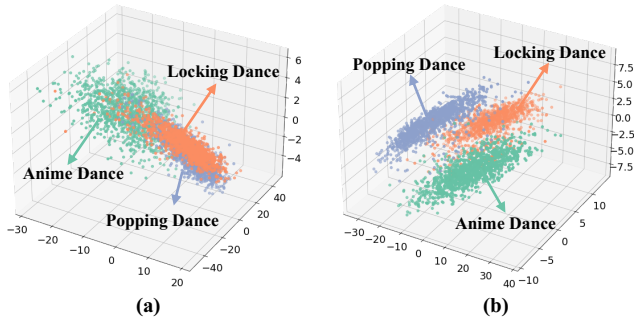

Figure 3: PCA-based visualization of style embeddings: (a) Original ones; (b) Enhanced by contrastive loss.

supervision on style embedding in the end-to-end learning of it, the inference between different style categories is obvious. To alleviate that, we employ a contrastive loss $\mathcal{L}_c$ to impose weak supervision on the style embedding leaner such that the learnt representations of styles $\{e^s\}$ will fall into separable clusters corresponding with different style categories. $\mathcal{L}_c$ is defined as:

$$\mathcal{L}_c(e^{s_i}, e^{s_j}) = \frac{1}{2}[(1-Y)(Dis)^2 + Y\{max(0, m - Dis)\}^2], \tag{2}$$

where (e^{s_i}, e^{s_j}) is a pair of style embeddings, and i, j are just used to identify the different style embeddings. Dis is the Euclidean distance between e^{s_i} and e^{s_j}. If a pair of style embeddings (e^{s_i}, e^{s_j}) are generated from the reference dances of an identical style, $Y = 0$. Otherwise $Y = 1$. m is the margin, set it to 2 to enforce the two examples of different styles far enough from each other.

Figure 3 (b) illustrates the style embeddings obtained under such loss function, where they fall into different clusters strongly coherent with their style labels, enabling us to perform style control more precisely in dance generation.

4.3 The Overall Music-to-dance Generator with Style Embedding

Since the relationship between music and dance is frame-aligned, we develop a music-to-dance generator module in an Auto-Encoder architecture for autoregressive dance generation. As described in Figure 2, in the music-to-dance generator, we apply a set of transformer layers as music encoders, and the dance generator is designed as an LSTM, followed by a fully connected layer.

Taking advantage of the special temporal structure composed of fully connected layers, the music encoder can obtain a long-range temporal representation from the input musical features $M \in \mathbb{R}^{T \times L}$. For notational simplicity, we ignore the temporal indicator t in the following. The final output of the encoders is the music representation $z_m \in \mathbb{R}^{T \times d_{model}}$, where d_{model} is the dimension of the music encoder's output, which should be equal to that of d_{gru} in the style embedding learner.

To generate dance with various initial poses, we add an extra pre-trained initial pose VAE component. We train this component on each dance pose in the known dance clips. We use a reconstruction L1 loss $\mathcal{L}_{init}^{rcon}$ and a KL loss $\mathcal{L}_{init}^{KL}$ to minimize reconstruction error after encoding and decoding:

$$\mathcal{L}_{init}^{rcon} = |\hat{d}^s - d^s|,$$
$$\mathcal{L}_{init}^{KL} = KL(\mathcal{N}(0, I) \| z_{init}), \tag{3}$$

where $KL(u\|v) = -\int u(z) \log \frac{u(z)}{v(z)} dz$, $d^s \in \mathbb{R}^{2J}$ represents the poses, and $J = 21$ indicates all the body joints. The initial pose VAE can enforce the dance pose into a latent distribution $Z_{init} \sim \mathcal{N}(0, I)$ via the KL loss.

After all, the dance generator receives a music representation z_m from the music encoder, an initial pose hidden vector (embedded by an FC layer) corresponding with an initial pose from VAE, and a style embedding e^s from the style embedding learner described in section 4.1. Then, the overall representation incorporating the 3 representations will be passed to the dance generator for dance generation. Finally, we supervise the generated dance $\hat{D}^s$ by using the reconstruction loss:

$$\mathcal{L}_G^{rcon} = |\hat{D}^s - D^s|, \tag{4}$$

Overall, we jointly train the style embedding learner and the music-to-dance generation model by optimizing the following objective $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_G^{rcon} + \lambda_{style}\mathcal{L}_c, \tag{5}$$

where λ_{style} is the weight of the related loss term.

5 Experiments

To evaluate our method, we conduct extensive experiments with the LSTM-based method proposed in (Shlizerman et al. 2018), DanceRevo (Huang et al. 2021), and Dancing2Music (Lee et al. 2019), which are the most representative ones for the dance generation task. We perform experiments from the following perspectives: Beat alignment between music and generated dance moves, style consistency within a continuous piece of generated dance moves, and diversity under the control of style embeddings, on both the I-Dance dataset and the AIST++ dataset (Li et al. 2021). Besides, we visualize the choreography by using a realistic human rendering model (Wang et al. 2018a; Chan et al. 2019) for a better intuitive evaluation. With regard to these different methods, we generated 16 dances (about 72k frames) for each dance style to calculate the quantitative scores. Code and supplementary video are available[1].

5.1 Implementation Details

In training, the style embedding learner comprises a stack of 3×3 kernel, 2×2 stride convolutional layers, followed by a 1-layers 128-unit GRU. We use 16, 32, 64 output channels for the convolutional stack. We apply a 4-head self-attention in this component and $W \in \mathbb{R}^{10 \times 128}$. For music-to-dance generation, we use 4 transformer layers in the music encoder with $d_{model} = 128$, and the dance generator is a 3-layer 128-unit LSTM with a fully connected layer (project 128 to 42). The initial pose VAE is the combination of two 3-layer dense layers, and the hidden dimensions are 42, 128, 16, 16, 128, and 42. The weight of the contrastive loss λ_{style} is set to 1.

[1] https://github.com/WilliammmZ/GenDance

Method	AIST++			I-Dance			User Prefer Ours %	Style Control
	Beat Align Score↑	Style$_c$ ↓	Style$_d$ ↑	Beat Align Score↑	Style$_c$ ↓	Style$_d$ ↑		
Real Dance	0.265	13.93	13.93	0.244	16.75	16.75	33%	-
LSTM (Shlizerman et al. 2018)	0.234	16.79	8.15	0.225	17.98	13.64	94%	✗
Dancing2Music (Lee et al. 2019)	0.221	**13.47**	9.70	**0.239**	17.91	14.13	61%	✗
DanceRevo (Huang et al. 2021)	0.238	14.01	11.50	0.202	17.61	**15.68**	55%	✗
Ours	**0.245**	14.06	**11.80**	0.230	**16.80**	14.73	-	✓

Table 1: Experiment results on I-Dance and AIST++ datasets for different dance generation methods.

5.2 Quantitative Evaluation

Beat alignment. Whether dance beats match music beats greatly affects the quality of the generated dance in terms of human perception. Here, we use the temporal distance between the music beat and its closest dance beat (Siyao et al. 2022) to evaluate how well they are matched, calculated as

$$\frac{1}{|B_m|}\sum_{t_m\in B_m}\exp\left\{-\frac{min_{t_d\in B_d}\|t_d-t_m\|^2}{2\sigma^2}\right\}, \quad (6)$$

where B_m and B_d are the sets of frame indexes (t_m and t_d) of music beats and dance beats, respectively. The music beats B_m can be easily extracted from music onset features by Librosa. For dance beats B_d, we mark each of such frames where the movement drastically slows down as a dance beat event (Ho et al. 2013). As shown in Table 1, our model achieves the best score on the AIST++ dataset and comparable performance to the Dancing2Music method on the I-Dance dataset. We are concerned with whether introducing style embedding to the encoder may interfere with the alignment between music beats and generative dance beats, but the experiment proves that our model is fully capable of achieving beat alignment.

Style consistency and diversity. In general, PCA projects the embedding to a subspace that preserves the discrimination power to the best extent, which holds also for this study. As shown in section 4.2, the style embeddings produced by our method fall into different clusters with a strong correlation to the style labels in PCA spanned space. So, we compute the euclidean distances between the style embeddings in PCA spanned space to measure style consistency and style diversity. Following the Fréchet Inception Distance (FID) (Heusel et al. 2017) used in (Lee et al. 2019; Siyao et al. 2022), the style distance measure is calculated as follows:

$$\frac{1}{N}\sum_{n=1}^{N}\{\frac{1}{M}\sum_{m=1}^{M}Dis[P(e^s_{j_n}),P(e^s_{k_m})]\}, \quad (7)$$

where $e^s_{j_n}$ and $e^s_{k_m}$ are the style embeddings produced for different dance examples of the same style s, while j and k denote whether the style embedding results from the coach dances or the generated ones. M and N are the number of the data examples. $P(\cdot)$ is the PCA operation. Let $e^s_{j_n}$ represent generated dance and $e^s_{k_m}$ coach dance. Then, Eq. (7) becomes style consistency score (style$_c$) to measure how well the generator is coached to approach the peered style. Provided both $e^s_{j_n}$ and $e^s_{k_m}$ correspond with generated dances, then, Eq. (7) becomes the style diversity score (style$_d$) to measure whether the generated dance only contains simply monotonous moves close to each other, just like repeating, or diverse motions comply with this style. As depicted in Table 1, our style consistency score is lower than those of the baselines on the I-Dance and closes to the runner-up method on the AIST++, which means that our model can learn the styles of coach videos more precisely, enabling finer control on the dance generator to approach desired styles. Our model also gains the best score on the AIST++ and a runner-up score on the I-Dance in terms of the diversity. The baselines model each style of dance individually, that is, training one model per dance style, but we use only a single model to learn the common patterns of all the categories of dances, which imposes a higher requirement on the generalizability of the model. It explains why baselines could perform better sometimes. In an overall sense, these results prove that employing the style embedding to guide generation can make the generated dance closer to the coach dance while ensure the diversity of dances within the same style category.

Method	Style$_c$ ↓	Style$_d$ ↑
Real Dance	13.93	13.93
Full Model	14.06	**11.80**
w.o. contrastive loss	**13.63**	9.59

Table 2: Result of ablation study on contrastive loss.

5.3 Ablation Study on Contrastive Loss

We conduct an ablation study on the contrastive loss for the style embedding learner, and the quantitative scores are shown in Table 2. We train two variant models without or with the contrastive loss on the AIST++ dataset. The "*w.o.* contrastive loss" gets the best score on Style$_c$. However, its score is even better than the real dance, which is abnormal in that only fewer dance moves are learnt from the coach videos, so the corresponding representations span a quite tiny set with more uniform patterns included, according to Eq. (7). At the same time, as indicated by the lower Style$_d$ score, without contrastive loss will lose diversity in the generated dances. As shown in Figure 3, imposing contrastive loss on style embedding learning makes the learnt patterns discriminative to be subject to style categories while main-

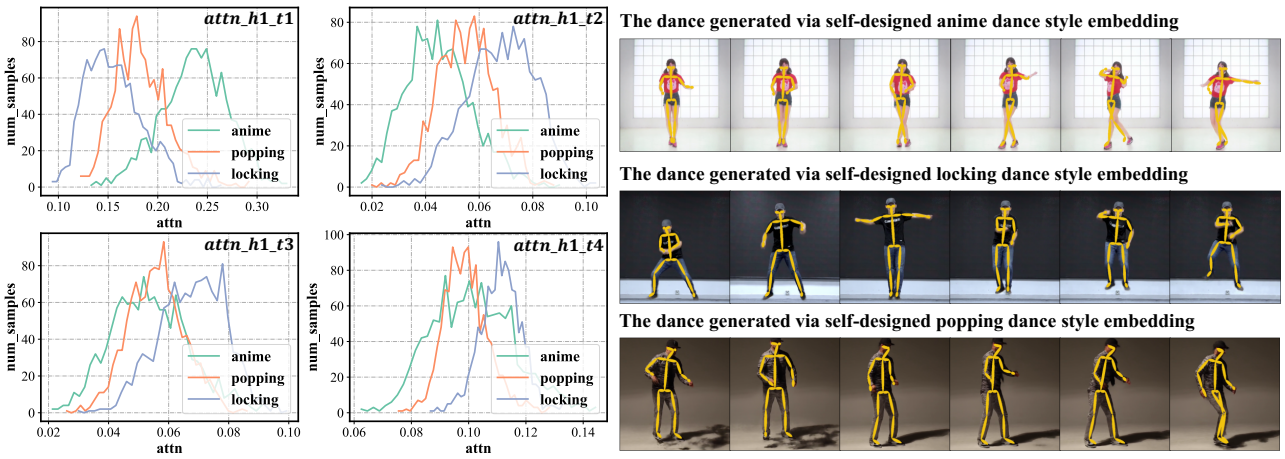

Figure 4: The left is the statistical distribution of the attention weights on different prototypes subject to 3 dance styles, where "h1" in "attn_h1_t1" means the first head of the multi-head self-attention and "t1" the first prototype in the codebook. The right displays the dances generated with the style embeddings tuned out from the statistical model in the left.

tains diversity within each category to prevent the learning from approaching a few monotonous patterns only.

5.4 Stylized Knowledge Applied to Dance Style Transfer and Control

The goal of this study is to mine the stylized dancing knowledge from real dance videos to direct style-concentrated dance generation. According to Figure 3, we have learnt style-discriminative representations of dance moves, where the learnt representations are not only style bearing but also diverse enough to composite the complex patterns of each style. As desired, the learnt representations in the form of style embedding fall into clusters without obvious overlap, nor converging to a few monotonous patterns in each cluster. Utilizing such knowledge learnt from coach dance videos, we can easily control the style of the generated dance by setting the values of style embeddings complying with a given style, that is, an inlier for the corresponding cluster. As described in section 4.1, the style embedding is based on the prototypes and the associated weights $attn$ to flag each dance style. To enable an intuitive insight into how the styles of generated dances are subject to the weights of prototypes, we illustrate in Figure 4 an example of the statistical distribution of the weights for each dance style as well as the dance moves generated with the manually set style embedding that comply with their statistical nature, which is modeled as Gaussian to govern the random setting for dance generation. According to Figure 4, the dances of different styles have different statistical nature to favor these prototype vectors differently. Hereby, we modify the style embedding in accordance with the corresponding statistical range of the attention weights to compose new dances as illustrated in Figure 4. In this way, our method can generate an infinite number of dance moves for a specific style accompanying the same song.

5.5 Qualitative Evaluation

Comparison to the existing methods. For qualitative evaluation, these dances are generated with LSTM (Ofli et al. 2008), DanceRevo (Huang et al. 2021), Dancing2Music (Lee, Lee, and Park 2013), and our method, trained on the I-Dance dataset. The visualizations are detailed in the supplementary video. Simply repeated rigid motions and nonstandard poses deviating remarkably from normal ones appear in the dances generated by the LSTM based model. Dancing2Music and DanceRevo avoid these problems by introducing the dance unit structure and the teacher-forcing scheme in their generation process, respectively. In contrast, the more realistic dance clips synthesized by our method demonstrate that the style embedding can also solve these problems.

User study. Following the previous work (Tang, Jia, and Mao 2018), we conduct a user study using a pairwise comparison scheme. We invite 18 students to complete this experiment. They are asked to select which one is realistic and preferred in a pair of generated dances using two different methods. The percentage of users who prefer the dance generated by our method is shown in Table 1. Comparing to the baselines, over 94%, 55%, and 61% of our generated dance is voted as the better one. However, compared with the real dance, there is still a gap. The percentage decreases to 33%.

6 Conclusion and Limitations

This paper proposes a novel style controllable dance generation framework that can learn stylized dancing knowledge from the reference dances and transfer them to the generated dances in an end-to-end manner. It introduces a new large-scale music-to-dance dataset. In terms of knowledge mining on dance move composition related to styles, we are at the very beginning to be aware of the 1-order knowledge figuring out how each prototype movement is favored by a style. In the future, we will explore high-order knowledge regarding the state transition from one prototype to another.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 62072469) and State Grid Corporation of China (Grant No. 5500-202011091A-0-0-00).

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.

Asahina, W.; Iwamoto, N.; Shum, H. P.; and Morishima, S. 2016. Automatic dance generation system considering sign language information. In *ACM SIGGRAPH 2016*, 1–2.

Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7291–7299.

Chan, C.; Ginosar, S.; Zhou, T.; and Efros, A. A. 2019. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*, 5933–5942.

Fan, R.; Xu, S.; and Geng, W. 2011. Example-based automatic music-driven conventional dance motion synthesis. *IEEE transactions on visualization and computer graphics*, 18(3): 501–515.

Ferreira, J. P.; Coutinho, T. M.; Gomes, T. L.; Neto, J. F.; Azevedo, R.; Martins, R.; and Nascimento, E. R. 2021. Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio. *Computers & Graphics*, 94: 11–21.

Gârbacea, C.; van den Oord, A.; Li, Y.; Lim, F. S.; Luebs, A.; Vinyals, O.; and Walters, T. C. 2019. Low bit-rate speech coding with VQ-VAE and a WaveNet decoder. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 735–739. IEEE.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Ho, C.; Tsai, W.-T.; Lin, K.-S.; and Chen, H. H. 2013. Extraction and alignment evaluation of motion beats for street dance. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2429–2433. IEEE.

Huang, R.; Hu, H.; Wu, W.; Sawada, K.; Zhang, M.; and Jiang, D. 2021. Dance Revolution: Long-Term Dance Generation with Music via Curriculum Learning. In *International Conference on Learning Representations*.

Kim, J. W.; Fouad, H.; Sibert, J. L.; and Hahn, J. K. 2009. Perceptually motivated automatic dance motion generation for music. *Computer Animation and Virtual Worlds*, 20(2-3): 375–384.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lee, H.-Y.; Yang, X.; Liu, M.-Y.; Wang, T.-C.; Lu, Y.-D.; Yang, M.-H.; and Kautz, J. 2019. Dancing to music. In *Advances in Neural Information Processing Systems*, 3586–3596.

Lee, M.; Lee, K.; and Park, J. 2013. Music similarity-based approach to generating dance motion sequence. *Multimedia tools and applications*, 62(3): 895–912.

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

McFee, B.; Raffel, C.; Liang, D.; Ellis, D. P.; McVicar, M.; Battenberg, E.; and Nieto, O. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 18–25.

Ofli, F.; Demir, Y.; Yemez, Y.; Erzin, E.; Tekalp, A. M.; Balcı, K.; Kızoğlu, İ.; Akarun, L.; Canton-Ferrer, C.; Tilmanne, J.; et al. 2008. An audio-driven dancing avatar. *Journal on Multimodal User Interfaces*, 2(2): 93–103.

Oord, A. v. d.; Vinyals, O.; and Kavukcuoglu, K. 2017. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*.

Peng, J.; Liu, D.; Xu, S.; and Li, H. 2021. Generating Diverse Structure for Image Inpainting With Hierarchical VQ-VAE. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10775–10784.

Razavi, A.; van den Oord, A.; and Vinyals, O. 2019. Generating diverse high-fidelity images with vq-vae-2. In *Advances in neural information processing systems*, 14866–14876.

Ren, X.; Li, H.; Huang, Z.; and Chen, Q. 2020. Self-supervised Dance Video Synthesis Conditioned on Music. In *Proceedings of the 28th ACM International Conference on Multimedia*, 46–54.

Shiratori, T.; Nakazawa, A.; and Ikeuchi, K. 2006. Dancing-to-music character animation. In *Computer Graphics Forum*, volume 25, 449–458. Wiley Online Library.

Shlizerman, E.; Dery, L.; Schoen, H.; and Kemelmacher-Shlizerman, I. 2018. Audio to body dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7574–7583.

Siyao, L.; Yu, W.; Gu, T.; Lin, C.; Wang, Q.; Qian, C.; Loy, C. C.; and Liu, Z. 2022. Bailando: 3D Dance Generation by Actor-Critic GPT with Choreographic Memory. *arXiv preprint arXiv:2203.13055*.

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.

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.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Tao, A.; Kautz, J.; and Catanzaro, B. 2018a. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Wang, Y.; Stanton, D.; Zhang, Y.; Ryan, R.-S.; Battenberg, E.; Shor, J.; Xiao, Y.; Jia, Y.; Ren, F.; and Saurous, R. A. 2018b. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, 5180–5189. PMLR.

Yalta, N.; Watanabe, S.; Nakadai, K.; and Ogata, T. 2019. Weakly-supervised deep recurrent neural networks for basic dance step generation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Yan, W.; Zhang, Y.; Abbeel, P.; and Srinivas, A. 2021. VideoGPT: Video Generation using VQ-VAE and Transformers. *arXiv preprint arXiv:2104.10157*.

Ye, Z.; Wu, H.; Jia, J.; Bu, Y.; Chen, W.; Meng, F.; and Wang, Y. 2020. ChoreoNet: Towards Music to Dance Synthesis with Choreographic Action Unit. In *Proceedings of the 28th ACM International Conference on Multimedia*, 744–752.