GroupDancer: Music to Multi-People Dance Synthesis with Style Collaboration

Zixuan Wang
wangzixu21@mails.tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China

Jia Jia
jjia@tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing National Research Center for Information Science and Technology
Beijing 100084, China

Haozhe Wu
wuhz19@mails.tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China

Junliang Xing*
jlxing@tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China

Jinghe Cai
caijh20@mails.tsinghua.edu.cn
Department of Computer Science and Technology, Tsinghua University
Beijing 100084, China

Fanbo Meng
fanbomeng@tencent.com
Tencent Technology Co., Ltd.

Guowen Chen
guowenchen@tencent.com
Tencent Technology Co., Ltd.

Yanfeng Wang
shawndwang@tencent.com
Tencent Technology Co., Ltd.

ABSTRACT

Different people dance in different styles. So when multiple people dance together, the phenomenon of style collaboration occurs: people need to seek common points while reserving differences in various dancing periods. Thus, we introduce a novel Music-driven Group Dance Synthesis task. Compared with single-people dance synthesis explored by most previous works, modeling the style collaboration phenomenon and choreographing for multiple people are more complicated and challenging. Moreover, the lack of sufficient records for conducting multi-people choreography in prior datasets further aggravates this problem. To address these issues, we construct a rich-annotated 3D Multi-Dancer Choreography dataset (MDC) and newly devise a metric SCEU for style collaboration evaluation. To our best knowledge, MDC is the first 3D dance dataset that collects both individual and collaborated music-dance pairs. Based on MDC, we present a novel framework, GroupDancer, consisting of three stages: Dancer Collaboration, Motion Choreography and Motion Transition. The Dancer Collaboration stage determines when and which dancers should collaborate their dancing styles from music. Afterward, the Motion Choreography stage produces a motion sequence for each dancer. Finally, the Motion Transition stage fills the gaps between the motions to achieve fluent and natural group dance. To make GroupDancer trainable from end to end and able to synthesize group dance with style collaboration, we propose mixed training and selective updating strategies. Comprehensive evaluations on the MDC dataset demonstrate that the proposed GroupDancer model can synthesize quite satisfactory group dance synthesis results with style collaboration.

CCS CONCEPTS

• **Applied computing** → **Media arts**.

KEYWORDS

Group Dance Synthesis, Choreography, Style Collaboration

ACM Reference Format:
Zixuan Wang, Jia Jia, Haozhe Wu, Junliang Xing, Jinghe Cai, Fanbo Meng, Guowen Chen, and Yanfeng Wang. 2022. GroupDancer: Music to Multi-People Dance Synthesis with Style Collaboration. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3503161.3548090

1 INTRODUCTION

An old saying goes, "To watch us dance is to hear our hearts speak." In the long process of human cultural and social development, dance has always been a basic form of art with individuality, consisting of personal feelings, motion habits, and music understanding. For individual dance, a single dancer usually gets much freedom to dance with his style. However, a group dance performed by multiple people simultaneously usually requires more style collaboration work for dancers to incorporate different ideas into the same music. As shown in Figure1, when professional dancers choreograph a piece of group dance, they devise various periods with different dancer activations based on their comprehension. Then the corresponding dancers collaborate to arrange motion sequences in every period.

*Corresponding authors

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3548090

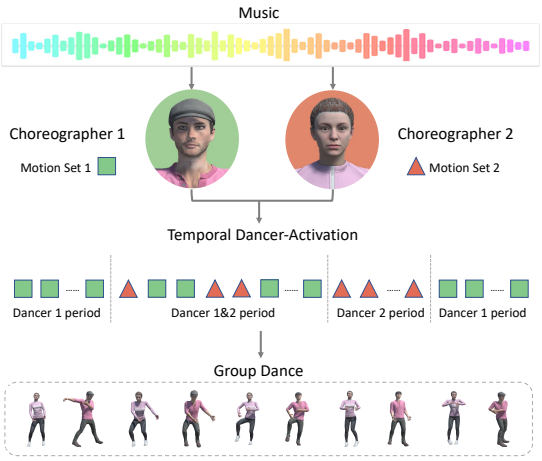

Figure 1: Multi-dancer group dance choreography procedure: firstly, determine various periods with different dancer activations and then arrange dance motions according to the corresponding dancers. It is noteworthy that the motions of different styles should be collaborated according to the music and each dancer's preference.

Extensive previous researches have shown the rationality of dances synthesis from music [4, 5, 8, 10, 14–18, 22, 25–27]. Early works [4, 10, 16] formulate music-to-dance synthesis as a similarity-based retrieval problem, which exhibits limited capacity. With the development of deep learning, recent works utilize deep learning models to achieve better consistency between music and generated dance [14, 15, 17, 18, 22, 25]. Besides, other recent works [5, 8, 26] leverage sequence prediction models to learn the mapping between music and dance phrases, considering the human choreography experience. Overall, these methods concentrate on improving the quality of individual dance with a single dancer; however, they ignore the task of synthesizing group dance with multiple dancers.

In this paper, we propose a novel problem called Music-driven Multi-Dancer Group Dance Synthesis, which aims to synthesize group dance according to the given music and the choreography preference of multiple dancers. Two main challenges below make this task tough to be accomplished:

– **Datasets Shortage.** Group dance synthesis needs dancer-wise choreography experience, while prior music-to-dance datasets seldom satisfy such requirements. AIST [23] released dancing videos, including group dances. However, further annotations are needed to put these videos into use. At the same time, group dance choreography in AIST suffers from too much easily dancing together rather than rational arrangements of different dancer activations. Additionally, a lack of dancer-wise choreography datasets causes problems in modeling the merging of dancing habits and choreography preferences via multi-dancer.
– **Lack of Choreography.** In human choreography procedure on group dance, dancers usually seldom have cooperation experience when they need to perform a group dance together. That is to say, collecting 3D dance data directly from group dance will have limited extensibility. Therefore, we need new methods and frameworks which can merge choreography habits and the experience of different dancers.

To address these issues, we construct a rich-annotated 3D Multi-Dancer Choreography dataset (MDC), which for the first time, contains both individual and collaborated music-dance pairs. We invite 10 dancers to annotate 725 3D dance motions from two dance types (Hiphop and Locking) in the first part. Additionally, each professional dancer is asked to arrange a motion phrase sequence for each music with 15 minutes of music (10~20 pieces, some shared). In the second part, we start by collecting 73 music pieces (60 minutes in total) from pop music. Afterward, we invite professional dancers to annotate the temporal dancer activation sequence for each music piece, just like the procedure in Figure 1. Besides, we newly devise a metric SCEU (style collaboration evaluation understudy) to measure the effect of style collaboration.

With the above dataset, we propose a three-stage music-driven group dance synthesis framework, GroupDancer, to imitate the human group dance choreography procedure. The GroupDancer consists of a Dancer Collaboration model, a Motion Choreography model, and a Motion Transition model: the Dancer Collaboration model learns the mapping from input music to dancer activation sequences; the Motion Choreography model infers the motion sequence of each dancer simultaneously according to the input music, and dancer activation sequences; and the Motion Transition model converts the motion sequences into continuous group dance motions. Figure 2 illustrates the deep architecture of our proposed GroupDancer framework. To make the deep architecture end-to-end trainable, we propose a novel loss function that helps the training process perform effective mixed training and selective updating. Specifically, for training data of individual music-dance pair, the proposed loss function takes corresponding motion results into account and update related network parameters. In this way, the generated results are restricted to the motions of specific dancers, and data from different dancers can be trained together.

Under the proposed framework, we implement an efficient system to perform group dance synthesis with style collaboration. We evaluate the system with extensive experimental analyses and comparisons on our MDC dataset for evaluation. Compared with baseline methods, our framework synthesizes more expressive and nature group dances. We conduct comprehensive user studies to investigate group dance diversity and dancer collaboration. With the mean opinion score (MOS) of 26 participants, our framework outperforms baseline methods by 1.08 on average in terms of group dance diversity and 1.81 on average in terms of dancer collaboration. Meanwhile, evaluation with SCEU demonstrates the efficiency of GroupDancer on style collaboration. To conclude, we summarize our contributions as follows:

- We introduce a novel Music-driven Multi-Dancer Group Dance Synthesis task, which aims to automatically synthesize group dance according to music and collaborate the styles of various dancers. To our best knowledge, this is the first work that proposes and works on such a problem.
- We construct a new rich-annotated MDC dataset, which contains 150 minutes of dancer-wise music-motion data with 10 dancers and 60 minutes of group dance choreography data with temporal dancer activation information. And we devise a new metric, SCEU, to measure the effect of style collaboration. To our best knowledge, MDC first collects individual and collaborated music-dance pairs.

- We imitate the human choreography procedure and devise a novel three-stage framework GroupDancer to produce multi-dancer group dance with style collaboration from given music. GroupDancer permits end-to-end and effective learning with a novel loss function and two new training strategies, including mixed training and selective updating.

2 RELATED WORK

Previous works related to our GroupDancer can be grouped into two aspects: music to dance synthesis and 3D dance dataset.

2.1 Music to Dance Synthesis

Music to dance synthesis has attracted many researchers to work on it. Most of them focus on achieving high consistency between music and generated dance and improving the creativity and diversity of their models. Early works generate dance motions following similarity-based retrieval methods [4, 10, 16], which usually result in unnatural transitions and limited capacity due to their static templates and rigid fashions. With the development of deep learning methods, more and more deep neural networks are exploited to extract the deeper relationship between music and dance. Crnkovic-Friis *et al.* [7] designed a Chor-RNN framework to predict dance motions which is the first deep learning-based method for this problem. Later, Tang *et al.* [22] utilized LSTM-autoencoder with L2 loss to synthesize 3D dance according to music. Huang *et al.* [12] proposed to use a curriculum learning strategy with L1 loss to alleviate error accumulation in long motion sequence generation. Wu *et al.* [25] implemented a dual learning framework using GANs for both music-to-dance and dance-to-music problems. AI Choreographer [18] devised a FACT model involving a deep cross-modal transformer block with full attention to generating realistic dance motions. In recent years, many works tried to imitate human dance production procedures. Choreonet [26] first defined choreographic dance units to fuse human choreography knowledge into the music-to-dance synthesis framework. DanceFormer [17] learned from the animation industry and formulated the music-to-dance task to predict critical poses and motion curves between them. However, all of these approaches focus on an individual dance, while group dance synthesis of multi-dancers receives little attention.

2.2 3D Dance Dataset

3D dance datasets play an essential role in the deep learning-based music-to-dance synthesis process. Unlike the 3D motion dataset, the 3D dance dataset usually requires much more professional experience and contains multi-modal information. Motion capture techniques are widely used in 3D dance dataset construction. Alemi *et al.* [1] released the first 3D dance dataset with synchronized music using motion capture data. While Tang *et al.* [22] invited professional dancers to construct a 3D mocap dataset of dance motions with music, which has some inevitable mismatches. Music2Dance [27] repaired the motion capture data according to the music and collected music-dance pair data with higher quality, but the fix process is labor-intensive. With the advance in 3D reconstruction methods, 3D dance data can be produced from 2D videos. These methods essentially reduce the time of data construction, although they will result in some losses of pose parameters accuracy. AI Choreographer [18] introduced the AIST++ dataset of 3D dance motions accompanied by music and multi-view images based on the 2D dance video database AIST [23]. ChoreoMaster [5] introduced a synchronized music and dance phrase dataset from both mocap resources and anime community resources. Still, the motions are all structured in four-beat meters, which may limit the capacity. DanceFormer [17] introduced the PhantomDance dataset labeled by experienced animators with parameters of critical poses and motion curves. However, the currently available 3D dance dataset all aims to deal with individual dance and can hardly be directly used for group dance. Although AIST [23] contains group dance videos, they still need more robust 3D reconstruction techniques for multi-human. To fill this gap, we invite professional dancers to construct a 3D Multi-Dancer Choreography dataset with individual and collaborated music-dance pairs.

3 PROBLEM FORMULATION

The problem setting of music-driven group dance synthesis is to predict the dance motions of multi-dancer from given music. Since group dance needs both dance motions choreography and dancers arrangement, there are two groups of predictive temporal attributes:

- **Dancer Activation Attributes**: da_t indicates which dancers will dance at time t.
- **Dance Motion Attributes**: For each dancer j, dm_t^j indicates which motion will be preformed at time t and $\mathbf{S}_r^j$, $\mathbf{S}_t^j$ means the joint-level rotations and translations at time t.

More intuitively, we formulate our problem as follows: Given acoustic features $\mathbf{A}_t$ of input music, we aim to synthesize sequences of human joint-level rotations $\mathbf{S}_r^j \in \mathbb{R}^{N\times J\times 4}$ and translations $\mathbf{S}_t^j \in \mathbb{R}^{N\times J\times 3}$, where j is the dancer index, N is the number of dance frames and J is the number of joints.

Thus, we proposed a three-stage music-driven group dance synthesis formulation considering human choreography experience. Formally:

- **Dancer Collaboration Stage**: In this stage, given input music, our objective is to learn a mapping from $\mathbf{A}_t, da_{t-1}$ to da_t, where da_t and $\mathbf{A}_t$ indicate dancer activation and acoustic features at time t.
- **Motion Choreography Stage**: Having obtained da_t, in this stage, we intend to learn a mapping from da_t, m_t and dm_{t-1}^j to dm_t^j, where dm_t^j denotes motion of dancer j at time t.
- **Motion Transition Stage**: In this stage, we aim to translate motion sequence $\{dm_1^j, \ldots, dm_t^j\}$ into joint-level rotations and translations, and fill the gap between adjacent motions. By now, the key results $(\mathbf{S}_r^j, \mathbf{S}_t^j)$ of group dance are obtained.

4 METHODOLOGY

Based on the formulation in Section 3, we propose a three-stage framework GroupDancer which imitates human choreography procedure to synthesize group dance, as shown in Figure 2. Our model takes a music piece as input and extracts acoustic features from it. Then, we feed the acoustic features into the Dancer Collaboration model to predict temporal dancer activation sequences. After that, we utilize the Multi-Dancer Motion Choreography model to generate a motion phrase sequence for each dancer. Finally, a Motion

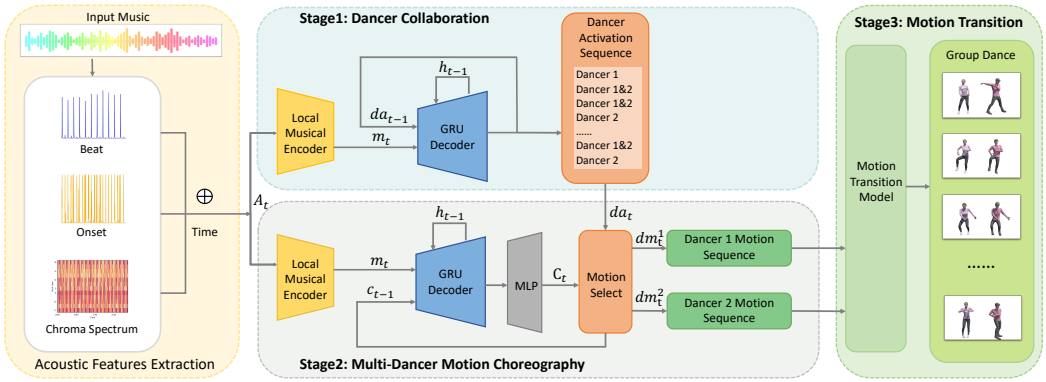

Figure 2: The overall workflow of GroupDancer consists of three stages. For the Dancer Collaboration Stage, we take the acoustic feature $\mathbf{A}_t$ as input and produce dancer activation da_t. Afterward, the Multi-Dancer Motion Choreography Stage combines $\mathbf{A}_t$ and da_t to predict dm_t^j, which indicates the motion of dancer j at time t. Eventually, the Motion Transition Model inpaints the transition gaps and generates natural and fluent group dance. Here we take the case of two dancers as an example. Extension to dancers of any number is also feasible. $\oplus$ denotes the concatenation operator.

Transition model is adopted to fill the gap between adjacent motions and synthesize complete group dance. In the following three subsections, we will go into detail about each model respectively.

4.1 Dancer Collaboration model

In the first stage of our framework, we propose a Dancer Collaboration model to predict dancer activation information, which determines when and which dancers should collaborate their dancing styles. Overall, our Dancer Collaboration model takes acoustic features as input and produces temporal dancer activation sequences. Details will be illustrated as follows.

For the input of Dancer Collaboration model, we extract acoustic features of the input music including Beat [3], Onset [9] and Chroma Spectrum [13] information with *Madmom* [2]. We represent the acoustic features respectively as $\mathbf{A}^{beat}$, $\mathbf{A}^{onset}$ and $\mathbf{A}^{chroma}$. Afterward, we concatenate the extracted features to formulate acoustic features, as shown on the left of Figure 2. Considering the temporal locality of music features, we exploit a sliding window to produce local music context. Specifically, given time t and a fixed slide window length w, we denote local acoustic features $\mathbf{A}_t$ as:

$$\mathbf{A}_t = \text{concat}(\mathbf{A}^{beat}_{[t-w/2,t+w/2]}, \mathbf{A}^{onset}_{[t-w/2,t+w/2]}, \mathbf{A}^{chroma}_{[t-w/2,t+w/2]}). \tag{1}$$

Having obtained the acoustic features, we now elaborate on the dancer activation synthesis process. We leverage a Local Musical Encoder and a GRU Decoder to predict dancer activation da_t from $\mathbf{A}_t$. Here da_t indicates at time t, which dancers should collaborate to dance with their styles mixed. For example, da_t = "Dancer1&2" means Dancer1 and Dancer2 dance together at time t. With the observation of human group dance, dancer collaboration usually changes between beats. Thus, we align $\{da_1, \ldots, da_t\}$ with beat sequence. Specifically, as shown in Figure2, the local acoustic features $\mathbf{A}_t$ are fed into the Local Musical Encoder to produce encoded musical feature m_t:

$$m_t = \text{encode}(\mathbf{A}_t). \tag{2}$$

As for the dancer activation prediction, we adopt a gated recurrent unit (GRU) [6] as our decoder which takes in m_t, da_{t-1} and h_{t-1}, and output $p(da_t)$ and h_t. Here h_t denotes the hidden state of the GRU at time t and $p(da_t)$ denotes the probability distribution of each dancer activation. And we select dancer activation with maximum probability in $p(da_t)$ as da_t. Thus, the functions of the GRU decoder are described as:

$$\begin{aligned} p(da_t), h_t &= \text{decode}(m_t, da_{t-1}, h_{t-1}), \\ da_t &= \text{argmax}(p(da_t)). \end{aligned} \tag{3}$$

Algorithm 1 Dancer Activation Generation

```
t ⟵ 1
DA_gen = []
da_{t-1} = StartOfDance
while da_{t-1} ≠ EndOfDance and the music is not ended do
    A_t = concat(A^beat_[t-w/2,t+w/2], A^onset_[t-w/2,t+w/2], A^chroma_[t-w/2,t+w/2])
    m_t = encode(A_t)
    p(da_t), h_t = decode(m_t, da_{t-1}, h_{t-1})
    da_t = argmax(p(da_t))
    Add da_t to DA_gen
    da_{t-1} ⟵ da_t
    t ⟵ t + 1
return DA_gen
```

It's worth noting, we create an ordered dancer activation set DA_n as defined domain of da_t according to the given dancers number n, which means $da_t \in \text{DA}_n$. Take $n = 2$ as an example, the DA_2 will be {"Dancer1", "Dancer2", "Dancer1&2"}.

In this way, the da_{t-1} and h_t contain the history of dancer activations and m_t embeds the acoustic features, which enable our model to fuse both dancer collaboration context and musical context. More details of the dancer activation generation algorithm are illustrated in the Algorithm 1, where DA_{gen} is the generated sequence of da_t.

During the training stage, we use expert annotations in the MDC dataset as ground truth. And we train the Dancer Collaboration model by minimizing the negative log-likelihood loss with the output $p(da_t)$ and the expert annotations da_t^{ground}:

$$\mathcal{L}_{nll}^{da} = -\sum_{t=1}^{N} \log(p(da_t = da_t^{ground})), \tag{4}$$

4.2 Multi-Dancer Motion Choreography model

This stage aims to predict motion sequences for each dancer in a group dance with the guidance of input music and dancer activation sequence produced in the last stage. Since both the music context and the motion history are required to synthesize the next motion, we leverage a seq2seq model to conduct multi-dancer choreography. Next, we will elaborate each part of this model in detail.

For the encoder of our Motion Choreography model, we leverage a Local Musical Encoder same as Section 4.1 and obtain encoded musical features m_t. Moreover, to model the time dependency and combine m_t with motion history, we utilize a GRU decoder and MLP layer to predict the next optional motions C_t, where C_t indicate motions of all the possible dancer activations. As shown in the Figure 2, all of these motion predictions share the same GRU, which means motion predictions of different dancer activations share the same hidden state. Therefore, at time t, we represent C_t as:

$$\begin{aligned} p(\mathrm{C}_t), h_t &= \mathrm{MLP}(\mathrm{decode}(m_t, c_{t-1}, h_{t-1})), \\ \mathrm{C}_t &= \mathrm{argmax}(p(\mathrm{C}_t)). \end{aligned} \tag{5}$$

Here h_t denotes the hidden state of the GRU and $p(\mathrm{C}_t)$ denotes the probability distribution of possible dance motions at time t.

We stress that C_t contains optional motions for all the dancer activations and the motion c_t we actually choose for the group dance at time t will be selected from C_t according to da_t. And we describe this selection process as:

$$c_t = \mathrm{Selection}(\mathrm{C}_t, da_t). \tag{6}$$

Having obtained c_t, we adopt da_t as a mask to generate the dance motion dm_t^j for individual dancer j at time t. In detail, if da_t says dancer j should dance at time t, then dancer j will perform c_t. Otherwise, dancer j should stop dance and wait for the next motion. Thus, we represent dm_t^j as:

$$dm_t^j = \mathrm{Mask}(j, c_t, da_t). \tag{7}$$

So far, we have synthesized motion sequences for multi-dancer of a group dance, and the Algorithm 2 shows the whole procedure of the motion generation. Here DM_j is generated sequence of dm_t^j.

For the training of the Multi-Dancer Motion Choreography model, we adopt two training strategies: mixed training and selective updating to endow the model with the capacity of style collaboration. Inspired by the human group dance procedure, mixed training is based on dancer-wise music-dance pairs data, which indicates each dancer's data contains only his motions. In short, each dancer's data contains a dancer's style, and mixed training will collaborate these styles. Detailed mixing schemes are described and compared in Section 6. Selective updating denotes that for each epoch, given training data of a specific dancer j, we select all the $c_t^i \in \mathrm{C}_t$ which satisfy $j \in \mathrm{DA}_n[i]$. And we just update MLP parameters related to these c_t^i. As a result, c_t^i are restricted to motions belonging to dancer j with $j \in \mathrm{DA}_n[i]$, which obey the rule of style collaboration. To implement these two strategies, we devise a selective negative log-likelihood loss $\mathcal{L}_{nll-sel}^{c}$:

$$\mathcal{L}_{nll-sel}^{c} = -\sum_{t=1}^{N} \sum_{j \in \mathrm{DA}_n[i]} \log(p(c_t^i = c_t^{ground})). \tag{8}$$

Algorithm 2 Multi-Dancer Motion Generation

$t \longleftarrow 1$
$n \longleftarrow DancerNumber$
for $j = 1$ to n **do**
  $\mathrm{DM}_j = []$
$c_{t-1} = StartOfDance$
while $c_{t-1} \neq EndOfDance$ **and** the music is not ended **do**
  $p(\mathrm{C}_t), h_t = \mathrm{MLP}(\mathrm{decode}(m_t, c_{t-1}, h_{t-1}))$
  $\mathrm{C}_t = \mathrm{argmax}(p(\mathrm{C}_t))$
  $c_t = \mathrm{Selection}(\mathrm{C}_t, da_t)$
  for $j = 1$ to n **do**
    $dm_t^j = \mathrm{Mask}(j, c_t, da_t)$
    Add dm_t^j to DM_j
  $c_{t-1} \longleftarrow c_t$
  $t \longleftarrow t + len(c_t)$
return $\{\mathrm{DM}_1, \ldots, \mathrm{DM}_n\}$

4.3 Motion Transition model

After the previous two stages, the dance motion sequence DM_j for each dancer j is generated. For this last stage, we devise a Motion Transition model to fill the gap between motion phrases and synthesize smooth and natural group dance. The Motion Transition model consists of three steps: 1) fill the motion gap, 2) align motion with music, and 3) group dance polish.

In more detail, inspired by QuaterNet [21], we represent our motion phrases dm_t^j using quaternion-based joint rotations and root translations. We first utilize Spherical linear interpolation (Slerp) for joint rotations inpainting and linear interpolation for root translations inpainting. Until now, we have obtained joint-level rotations $\mathrm{S}_r^j \in \mathbb{R}^{N \times J \times 4}$ and translations $\mathrm{S}_t^j \in \mathbb{R}^{N \times J \times 3}$.

Afterward, we align S_r^j and S_t^j with musical beats. For more accurate kinematic beats control, we ask the professional dancers to annotate the kinematic beats of their motion phrases. This will also maintain the individuality of each dancer's motion style.

Moreover, we conduct necessary polish work automatically for the synthesized music-dance pairs to improve the naturalness and completeness of the group dance. Based on our observations of real dance, we find that for the "*Hold*" motion, the dancers will keep the pose of the last motion and also have breathing movements. Thus, we adopt a Deterministic Finite Automaton to implement the breathing effect referring to the game industry. Besides, for group dances like Hiphop and Locking, the dancers usually randomly dance some freestyle before the dance routine starts. Thus, we invite professional dancers to annotate their frequently used pre-dance freestyle. Based on this, we synthesize these pre-dance freestyles randomly. So far, we have produced a complete group dance.

5 DATASET

In this section, we exhaustively introduce our MDC dataset. We also conduct a comparison to some similar datasets which are available currently, as shown in Table 1.

Since that both dancers' arrangement choreography and dance motions choreography are significant to the group dance, we construct our MDC dataset in the following two parts: Dancer Choreography and Dancer-wise Motion Choreography. For the first part, we collect music-dance pairs for group dances and annotate dancer activation information to learn the mapping between music and dancer choreography. For the second part, we collect music-dance pairs from individual dances and annotate motion information to learn the motion preferences and dance styles of different dancers. The reason we do not use group dance in the second part is that individual dance data better reveals the style of each dancer, and also, group dance data can not be applied to new dancer combinations.

5.1 Dancer Choreography

In this part, our aim is to learn the knowledge of dancer choreography of group dance.

Music Acquisition: Group dance has been largely ignored in previous dance datasets. To our best knowledge, only AIST[23] recorded some group dance videos. However, the entire time is still short. The music of Hiphop group dance with cooperation only sums up 8.22 minutes in AIST, and most of the choreography is just dancing together, which can not reveal the changeability of group dance. To address these issues, we invite choreographers to select 60 minutes of group dance videos from Youtube, Bilibili, and AIST[23], in which dance genres vary from Hiphop to Locking. Then, we extract music pieces from the videos.

Dancer Activation Annotation: Having obtained the music, we ask the choreographers to annotate the dancer activation sequence with time information for each music piece, considering the choreography of the corresponding video. In total, we obtained 73 collaborated music-dance pairs, which sum up 60 minutes. By now, we have completed the first stage of the MDC dataset.

5.2 Dancer-wise Motion Choreography:

In this part, we intend to construct a dancer-wise motion choreography dataset that can reveal the motion preference and dancing habits of each dancer, respectively.

Music Acquisition: We select dance videos with Hiphop, Locking and Pop music from Youtube and Bilibili, and then extract the audio from the videos. In totall, we acquire 96.52 minutes of music, among which there are 85 pieces of music.

Motion Annotation: To better record the individuality of dancers, we invite 10 dancers of Hiphop and Locking to build 10 dancer-wise motion sets for themselves, respectively. Initially, we collect raw dance motion videos from YouTube, Bilibili, and AIST[23], in which there are 45 Hiphop motions and 27 Locking motions. Then we reconstruct 3D motions (fbx file) from these motion videos and we give each dancer motion files of his genre. Then we ask each dancer to scan the given motion fbx files and mark the motions in the files. The necessary motion attributes are "Start frame", "End frame", "Beat number" and "Beat frames". Finally, we construct a motion set for each dancer and in all we got 725 dance motions.

Choreography Annotation: We firstly ask each dancer to choose 15 minutes of music from all 85 pieces of music. Then, the dancers are told to choreograph every music piece they have chosen. In detail, they are required to arrange a motion sequence for each music and mark the start time and end time of each motion with their own motion set from the former part. Finally, we collect 165 individual music-dance pairs.

6 EXPERIMENT

In this section, we conduct extensive experiments to demonstrate the effectiveness of our framework. We evaluate our framework on the collected MDC dataset. Our method has acquired better synthesis results both qualitatively and quantitatively.

6.1 Implementation Details

Since prior works didn't publish 3D datasets for the group dance, we only conduct experiments on the MDC dataset we built. Section 5 has introduced the construction of the MDC dataset, in which we totally collect 73 dancer-wise music-dance pairs and we select 10 pieces for testing and hold out the remained 63 pieces for training.

Before training, We first use *Madmom* [2] to extract acoustic features from the input music. For the beat feature $\mathbf{A}^{beat}$, onset feature $\mathbf{A}^{onset}$ and deep chroma spectrum feature $\mathbf{A}^{chroma}$, the frame-per-second (FPS) are 100, 100, 10 respectively. Then in the local musical encoder, $\mathbf{A}^{beat}$ and $\mathbf{A}^{onset}$ are convolved through 3 convolutional layers, $\mathbf{A}^{chroma}$ is convolved through 5 convolutional layers. Then they are fed to an MLP layer to output local musical features. Afterward, concatenating these features results in a 64-dimension feature vector $\mathbf{A}_t$ at time t. For the GRU decoder, the embedding dimension is set to the number of possible options.

As for the training of Dancer Collaboration and Multi-Dancer Motion Choreography models, we adopt RMSprop algorithm [11] with an initial learning rate at 10^{-3}. Besides, we apply ReduceOnPlateu learning rate decay strategy from PyTorch [20] with patience set to 8, the decay factor to 0.9. To accelerate the training process, we adopt teacher-forcing [24] technique with a probability of 0.3.

6.2 Metrics

We evaluate our framework with the following different metrics:

SCEU: BLEU [19] is commonly used in machine-translation and music-to-dance problems. However, it can hardly reflect the effectiveness of style collaboration because imbalanced collaboration and good matching to references both result in high BLEU value, which are conflicted in style collaboration. Thus, we introduce a novel metric SCEU (style collaboration evaluation understudy) to evaluate the style collaboration effectiveness. As shown in Equation (9), we change the arithmetic mean of BLEU into the geometric mean for SCEU. Here *NC* indicates the number of candidates which means the synthesized sequences, while NR indicates the number of references which means the ground truth sequences. In detail, we will explain the specific usage of SCEU in Section6.3.

$$\text{SCEU}_n = \frac{\sum_{c\in candi}[\prod_{c'\in ref}\frac{\sum_{n-gram\in c}\text{Count}(n-gram)}{\sum_{n-gram'\in c'}\text{Count}(n-gram')}]^{1/NR}}{NC}. \quad (9)$$

Dataset	Group dance	multi-dancer	3D $Joint_{params}$	motion phrase	motion beat annotations
Dance with Melody	X	X	✓	✓	X
AIST	✓	✓	X	X	X
AIST++	X	✓	✓	X	X
ChoreoMaster	X	X	✓	✓	X
MDC	✓	✓	✓	✓	✓

Table 1: Our MDC dataset *v.s.* the other similar datasets, namely Dance with Melody[22], AIST[23], AIST++[18] and ChoreoMaster[5]. To our best knowledge, MDC is the first rich-annotated 3D dance dataset, which contains both dancer-wise individual music-dance pairs and collaborated group dance music-dance pairs.

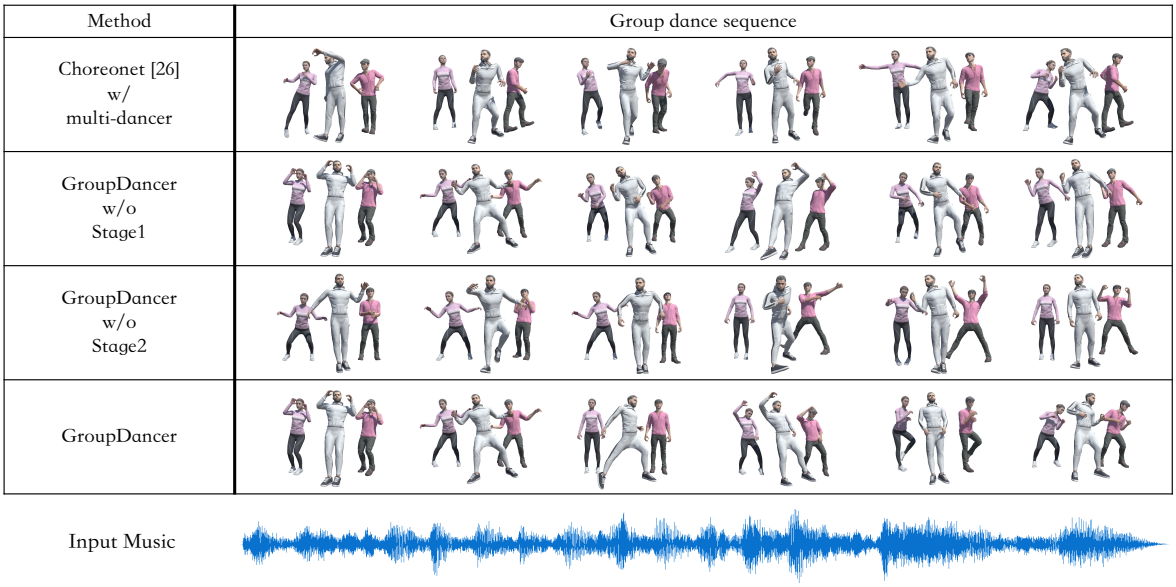

Figure 3: Visual comparison of group dance sequences synthesized by our GroupDancer and baseline methods. The dances generated by Choreonet and GroupDancer w/o Stage2 show rare collaborations. The dance generated by GroupDancer w/o Stage1 falls in easily dancing the same. While GroupDancer produces dance with both dance collaboration and individual style.

Besides, we conduct extensive user studies to evaluate the quality of the synthesis results. Participants are required to rate the following factors from 0 to 5: (1) The diversity of dance choreography and motions. (2) The naturalness of group dance motions and transitions. (3) The matchness between dance and music. (4) The Collaboration of dancers in the group dance.

Table 2: Comparison of Different Methods with mean of $SCEU_n$.

Mixed Strategy	$SCEU_1 \uparrow$	$SCEU_2 \uparrow$	$SCEU_3 \uparrow$	$SCEU_4 \uparrow$
DBD	**0.2968**	**0.1957**	**0.0954**	**0.0741**
PBP	0.2715	0.1744	0.0687	0.0461
RS	0.2947	0.1955	0.0903	0.0596

6.3 Comparison on Mixed Training

In this section, we compare different mixed training strategies on our Multi-Dancer Motion Choreography stage. In fact, the training process of this stage aims to solve a problem with a trade-off between music-dance relations and dancer-motion relations. That's to say, when given a piece of music and dance motion history, the next motion is influenced by the previous motion and the local music features. The two worst groups of results are: (1) totally depend on music-dance relations; (2) totally depend on dancer-motion relations. To avoid the two groups, we propose **DBD** mixed training method in which we arrange the music-dance pairs dancer by dancer, so that each dancer's style will be better collaborated.

Overall, we compared our **DBD** mixed strategy with the following methods: (1) **PBP:** arrange the music-dance pairs piece by piece; (2) **RS:** arrange the music-dance pairs randomly. To evaluate the effectiveness of these mixed training strategies, we devise SCEU, as shown in Equation (9). Higher SCEU values reveal a more balanced proportion of styles which indicates the model learns the style collaboration better. Each time we train a model for NC dancers, we prepare NR music, which is shared by these NC dancers. Thus, we obtain $NC * NR$ music-dance pairs as the test set.

The results are shown in Table 2. From the comparison, we observe that our **DBD** method has acquired higher style collaboration. At the same time, **RS** method also obtains good results, however the Figure 4 shows the variance of **RS** results is about two times

bigger than **DBD**. Hence, our **DBD** mixed strategy is more stable and better helps the model endow the style collaborating capacity.

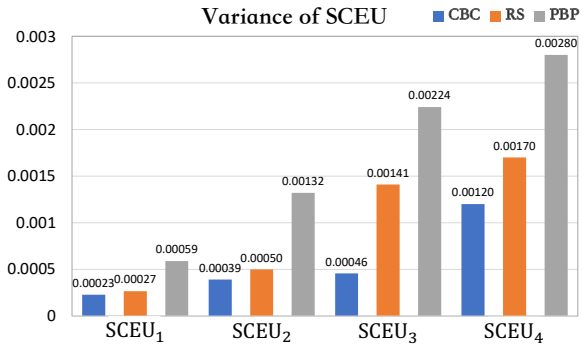

Figure 4: Visualization of the SCEU variance for comparison of different mixed training methods. This figure shows that the CBC method outputs the most stable results while the RS method leads to more bias.

6.4 Comparison with Individual Dance Model

In this section, we conduct experiments to demonstrate that our method synthesizes more natural and expressive group dance compared with the individual dance models. Specifically, we compare our GroupDancer with the following baseline methods: (1) the Choreonet [26] framework with single-dancer, (2) the Choreonet framework with multi-dancer, (3) the ChoreoMaster [5] framework with single-dancer, (4) GroupDancer without stage1, (5) GroupDancer without stage2. Since Choreonet can only synthesize dances with single dancer in each prediction, for (2) we input the same music and repeat the synthesis process to produce dance with multiple dancers. For (3), we use the dataset published by ChoreoMaster since the format of training data is different from our MDC dataset. For (4) and (5), we aim to verify the significance of the first and second stages in our framework design. We showcase some generated group dances in Figure 3, which show the effectiveness of our GroupDancer framework.

To further investigate the quality of synthesized dances, we invite 26 participants to rate dance diversity and dancer collaboration with the mean opinion score (MOS) in the range of 1-5. As shown in Table 3, our GroupDancer outperforms baseline methods by 1.08 on average in terms of dance diversity and 1.81 on average in terms of dancer collaboration. Meanwhile, the results demonstrate that both stage 1 and stage 2 are crucial to our framework.

Table 3: Comparison on Ours GroupDancer and individual dance models.

Methods	Diversity	Dancer Collaboration
Ours	**4.46**	**4.08**
Ours w/o Stage1	3.27	2.27
Ours w/o Stage2	3.08	1.77
ChoreoNet w/ multi-dancer	3.38	1.73
ChoreoNet w/ single-dancer	2.85	-
ChoreoMaster w/ single-dancer	2.96	-

6.5 Ablation Study

In this section, we conduct an ablation study on the Motion Transition Stage which is the third stage of our framework. Overall, we compare our Motion Transition model with the following methods: (1) Ours w/o pre-dance. This method ablates the motions for the prelude. (2) Ours w/o breath effect. This method replaces the slight breathing with staying still when dancers don't dance. (3) Ours w/ perturbation. This method add some small perturbations to the dancers' motion sequences to simulate the irregularities that may occur in human dance. To evaluate these methods, we conduct a user study. 26 participants are asked to rate the dance naturalness and music matchness of synthesized dances. As shown in Table 4, the results demonstrate that the design of pre-dance and breath effect all increase the performance of our framework by a large margin and the perturbation results in an adverse impact on the group dance. Some feedback from participants says the perturbation greatly decreases the fluency so most people think it's unnatural.

Table 4: Comparison on Ours Motion Transition model.

Methods	Dance Naturalness	Music Matchness
Ours	**4.04**	**4.27**
Ours w/o pre-dance	2.46	2.96
Ours w/o breath effect	3.35	3.77
Ours w/ perturbation	2.27	2.73

7 CONCLUSION

In this paper, we propose a novel task of Music-driven Multi-dancer Group dance Synthesis. According to real human choreography experience, we first formulate this problem as a three-stage procedure. Moreover, to fill in gaps of group dance choreography, we construct an MDC dataset that consists of both individual and collaborated music-dance pairs. Besides, we newly devise a metric SCEU for style collaboration evaluation. Based on the MDC dataset, we imitate the human choreography procedure and devise a three-stage framework GroupDancer to automatically generate group dance with style collaboration from input music. To make the deep architecture end-to-end trainable, we propose a novel loss function $\mathcal{L}^{c}_{nll-sel}$ which helps the training process perform effective mixed training and selective updating. Additionally, we conduct extensive experiments on the MDC dataset and obtain expressive synthesis results with our GroupDancer model. The constructed MDC dataset will be made publicly available in the future. We hope that the proposal of style collaboration and the construction of the MDC dataset pave a new way for music-driven group dance synthesis.

8 ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China under Grant No.2021QY1500, the state key program of the National Natural Science Foundation of China (NSFC) (No.61831022 and No.62076238) and Tiangong Institute for Intelligent Computing, Tsinghua University. We would like to sincerely thank Shuoyi Zhou from Tsinghua University and Rui Niu from Beijing University of Technology for their support to the work.

REFERENCES

[1] Omid Alemi, Jules Françoise, and Philippe Pasquier. 2017. GrooveNet: Real-time music-driven dance movement generation using artificial neural networks. *Workshop on Machine Learning for Creativity, ACM SIGKDD Conference on Knowledge Discovery and Data Mining* 8, 17 (2017), 26.

[2] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. 2016. Madmom: A new python audio and music signal processing library. In *Proceedings of the ACM International Conference on Multimedia (MM)*. 1174–1178.

[3] Sebastian Böck and Markus Schedl. 2011. Enhanced beat tracking with context-aware neural networks. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*. 135–139.

[4] Marc Cardle, Loic Barthe, Stephen Brooks, and Peter Robinson. 2002. Music-driven motion editing: Local motion transformations guided by music analysis. In *Proceedings of the Eurographics UK Conference (EGUK)*. 38–44.

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

[6] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.

[7] Luka Crnkovic-Friis and Louise Crnkovic-Friis. 2016. Generative Choreography using Deep Learning. In *Proceedings of the International Conference on Computational Creativity (ICCC)*.

[8] Yinglin Duan, Tianyang Shi, Zhengxia Zou, Jia Qin, Yifei Zhao, Yi Yuan, Jie Hou, Xiang Wen, and Changjie Fan. 2020. Semi-supervised learning for in-game expert-level music-to-dance translation. *arXiv preprint arXiv:2009.12763* (2020).

[9] Florian Eyben, Sebastian Böck, Björn Schuller, and Alex Graves. 2010. Universal onset detection with bidirectional long-short term memory neural networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 589–594.

[10] Rukun Fan, Songhua Xu, and Weidong Geng. 2011. Example-based automatic music-driven conventional dance motion synthesis. *Transactions on Visualization and Computer Graphics (TVCG)* 18, 3 (2011), 501–515.

[11] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850* (2013).

[12] Ruozi Huang, Huang Hu, Wei Wu, Kei Sawada, Mi Zhang, and Daxin Jiang. 2020. Dance Revolution: Long-Term Dance Generation with Music via Curriculum Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

[13] Filip Korzeniowski and Gerhard Widmer. 2016. Feature learning for chord recognition: The deep chroma extractor. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 37–43.

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

[15] Juheon Lee, Seohyun Kim, and Kyogu Lee. 2018. Listen to dance: Music-driven choreography generation using autoregressive encoder-decoder network. *arXiv preprint arXiv:1811.00818* (2018).

[16] Minho Lee, Kyogu Lee, and Jaeheung Park. 2013. Music similarity-based approach to generating dance motion sequence. *Multimedia Tools and Applications* 62, 3 (2013), 895–912.

[17] Buyu Li, Yongchi Zhao, Shi Zhelun, and Lu Sheng. 2022. Danceformer: Music conditioned 3d dance generation with parametric motion transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1272–1279.

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

[19] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*. 311–318.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems (NIPS)* 32 (2019).

[21] Dario Pavllo, David Grangier, and Michael Auli. 2018. QuaterNet: A Quaternion-based Recurrent Model for Human Motion. In *Proceedings of the British Machine Vision Conference (BMVC)*. 1–18.

[22] Taoran Tang, Jia Jia, and Hanyang Mao. 2018. Dance with melody: An lstm-autoencoder approach to music-oriented dance synthesis. In *Proceedings of the ACM International Conference on Multimedia (MM)*. 1598–1606.

[23] Shuhei Tsuchida, Satoru Fukayama, Masahiro Hamasaki, and Masataka Goto. 2019. AIST Dance Video Database: Multi-genre, Multi-dancer, and Multi-camera Database for Dance Information Processing. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. Delft, Netherlands, 501–510.

[24] Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation* 1, 2 (1989), 270–280.

[25] Shuang Wu, Zhenguang Liu, Shijian Lu, and Li Cheng. 2021. Dual Learning Music Composition and Dance Choreography. In *Proceedings of the ACM International Conference on Multimedia (MM)*. 3746–3754.

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

[27] Wenlin Zhuang, Congyi Wang, Jinxiang Chai, Yangang Wang, Ming Shao, and Siyu Xia. 2022. Music2Dance: DanceNet for Music-Driven Dance Generation. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 18, 2 (2022), 1–21.