DANCE REVOLUTION: LONG-TERM DANCE GENERATION WITH MUSIC VIA CURRICULUM LEARNING

Ruozi Huang[1*†] **Huang Hu**[2*] **Wei Wu**[3] **Kei Sawada**[4] **Mi Zhang**[1] **Daxin Jiang**[2‡]
Fudan University[1] Microsoft STCA[2] Meituan[3] Rinna AI[4]
{rzhuang17,mi_zhang}@fudan.edu.cn, {huahu,djiang}@microsoft.com,
wuwei19850318@gmail.com, keisawada@rinna.co.jp

ABSTRACT

Dancing to music is one of human's innate abilities since ancient times. In machine learning research, however, synthesizing dance movements from music is a challenging problem. Recently, researchers synthesize human motion sequences through autoregressive models like recurrent neural network (RNN). Such an approach often generates short sequences due to an accumulation of prediction errors that are fed back into the neural network. This problem becomes even more severe in the long motion sequence generation. Besides, the consistency between dance and music in terms of style, rhythm and beat is yet to be taken into account during modeling. In this paper, we formalize the music-conditioned dance generation as a sequence-to-sequence learning problem and devise a novel seq2seq architecture to efficiently process long sequences of music features and capture the fine-grained correspondence between music and dance. Furthermore, we propose a novel curriculum learning strategy to alleviate error accumulation of autoregressive models in long motion sequence generation, which gently changes the training process from a fully guided teacher-forcing scheme using the previous ground-truth movements, towards a less guided autoregressive scheme mostly using the generated movements instead. Extensive experiments show that our approach significantly outperforms the existing state-of-the-arts on automatic metrics and human evaluation. We also make a demo video to demonstrate the superior performance of our proposed approach at https://www.youtube.com/watch?v=lmE20MEheZ8.

1 INTRODUCTION

Arguably, dancing to music is one of human's innate abilities, as we can spontaneously sway along with the tempo of music we hear. The research in neuropsychology indicates that our brain is hardwired to make us move and synchronize with music regardless of our intention (Chen et al., 2008). Another study of archaeology also suggests that dance is a social communication skill among early humans connected to the ability of survival long time ago (Adshead-Lansdale & Layson, 2006). Nowadays, dance (to music) has become a means to the cultural promotion, a method of emotional expression, a tool for socialization and an art form to bring aesthetic enjoyment. The neurological mechanism behind dancing behavior and the unique value of dance to society motivate us to explore a computational approach to dance creation from a piece of music in artificial intelligence research. Such work is potentially beneficial to a wide range of applications such as dance creation assistant in art and sports, character motion generation for audio games and research on cross-modal behavior.

In literature, the music-conditioned dance generation is a relatively new task that attracts increasing research interests recently. Early works (Fan et al., 2011; Lee et al., 2013) synthesize dance sequences from music by retrieval-based methods, which show the limited creativity in practice. Recently, Lee et al. (2019) formulate the task from the generative perspective, and further propose a decomposition-to-composition framework. Their model first generates basic dance units from the music clips and then composes them by using the last pose of current unit to initialize the first pose of the next

*Equal contribution.
†Work done during the internship at Microsoft STCA.
‡Corresponding author.

unit. Although this approach shows better performance than the retrieval-based methods, several challenges still remain.

First, existing generative methods synthesize new human motion sequences through autoregressive models like RNN, which tend to result in short sequences. In other words, the generated sequences often quickly freeze within a few seconds due to an accumulation of prediction errors that are fed back into the neural network. This problem becomes even more severe in long motion sequence generation. For instance, composing a dance for a 1-minute music clip under 15 Frame Per Second (FPS) means generating 900 poses at one time. In practice, we need novel methods which can effectively generate long motion sequences. Besides, how to enhance the harmony between the synthesized dance movements and the given music is a largely unexplored challenge. Inutitively, the movements need to be consistent with the music in terms of style, rhythm and beat. However, to achieve this goal is non-trivial, which requires the generation model to have the capability to capture the fine-grained correspondence between music and dance.

In this paper, we formalize music-conditioned dance generation as a sequence-to-sequence learning problem where the fine-grained correspondence between music and dance is represented through sequence modeling and their alignment is established via mapping from a sequence of acoustic features of the music to a sequence of movements of the dance. The model consists of a music encoder and a dance decoder. The encoder transforms low-level acoustic features of an input music clip into high-level latent representations via self-attention with a receptive field restricted within k-nearest neighbors of an element. Thus, the encoder can efficiently process long sequences of music features, e.g., a sequence with more than 1000 elements, and model local characteristics of the music such as chord and rhythm patterns. The decoder exploits a recurrent structure to predict the dance movement frame by frame conditioned on the corresponding element in the latent representations of music feature sequence. Furthermore, we propose a curriculum learning (Bengio et al., 2009) strategy to alleviate error accumulation (Li et al., 2017) of autoregressive models in long motion sequence generation. Specifically, it gently changes the training process from a fully guided teacher-forcing scheme using the previous ground-truth movements, towards a less guided autoregressive scheme which mostly utilizes the generated movements instead. This strategy bridges the gap between training and inference of autoregressive models, and thus alleviates error accumulation at inference.

The length of each video clip in the dataset released by Lee et al. (2019) is only about 6 seconds, which cannot be used by other methds to generate long-term dance except for their specially designed decomposition-to-composition framework. To facilitate the task of long-term dance generation with music, we collect a high-quality dataset consisting of 1-minute video clips, totaling about 12 hours. And there are three representative styles in our dataset: “Ballet”, “Hiphop” and “Japanese Pop”.

Our contributions in this work are four-fold: (1) We formalize music-conditioned dance generation as a sequence-to-sequence learning problem and devise a novel seq2seq architecture for the long-term dance generation with music. (2) We propose a novel curriculum learning strategy to alleviate error accumulation of autoregressive models in long motion sequence generation. (3) To facilitate long-term dance generation with music, we collect a high-quality dataset that is available with our code[1]. (4) The extensive experiments show that our approach significantly outperforms the existing state-of-the-arts on both automatic metrics and human judgements. The demo video in the supplementary material also exhibits our approach can generate diverse minute-length dances that are smooth, natural-looking, style-consistent and beat-matching with the musics from test set.

2 Related Work

Cross-Modal Learning. Most existing works focus on the modeling between vision and text, such as image captioning (Lu et al., 2017; Xu et al., 2015) and text-to-image generation (Reed et al., 2016; Zhang et al., 2017). There are some other works to study the translation between audio and text like Automatic Speech Recognition (ASR) (Hinton et al., 2012) and Text-To-Speech (TTS) (Oord et al., 2016). While the modeling between audio and vision is largely unexplored and the music-conditioned dance generation is a typical cross-modal learning problem from audio to vision.

[1] `https://github.com/stonyhu/DanceRevolution`

Human Motion Prediction. Prediction of human motion dynamics has been a challenging problem in computer vision, which suffers from the high spatial-temporal complexity. Existing works (Chan et al., 2019; Wang et al., 2018) represent the human pose as 2D or 3D body keyjoints (Cao et al., 2017) and address the problem via sequence modeling. Early methods, such as hidden markov models (Lehrmann et al., 2014), Gaussian processes (Wang et al., 2006) and restricted boltzmann machines (Taylor et al., 2007), have to balance the model capacity and inference complexity due to complicated training procedures. Recently, neural networks dominate the human motion modeling. For instance, Fragkiadaki et al. (2015) present LSTM-3LR and Encoder-Recurrent-Decoder (ERD) as two recurrent architectures for the task; Jain et al. (2016) propose a structural-RNN to model human-object interactions in a spatio-temporal graph; and Ghosh et al. (2017) equip LSTM-3LR with a dropout autoencoder to enhance the long-term prediction. Besides, convolutional neural networks (CNNs) have also been utilized to model the human motion prediction (Li et al., 2018).

Audio-Conditioned Dance Generation. In the research of audio-conditioned dance generation, most existing works study 2D dance motion generation with music since the training data for paired 2D pose and music can be extracted from the huge amount of dance videos available online. Various methods have been proposed to handle this task, such as adversarial learning based methods (Lee et al., 2019; Sun et al., 2020; Ferreira et al., 2021), autoencoder methods (Tang et al., 2018) and sequence-to-sequence methods (Lee et al., 2018; Ren et al., 2019; Yalta et al., 2019; Ye et al., 2020). While these works mainly focus on exploring the different neural architectures and overlook the freezing motion issue in dance motion synthesis. In this work, we first propose a novel seq2seq architecture to model the fine-grained correspondence between music and dance, and then introduce a novel curriculum learning strategy to address the freezing motion issue caused by error accumulation (Li et al., 2017) in long-term dance motion generation.

3 Approach

In this section, we present our approach to music-conditioned dance generation. After formalization of the problem in question, we elaborate the model architecture and the dynamic auto-condition learning approach that facilitates long-term dance generation according to the given music.

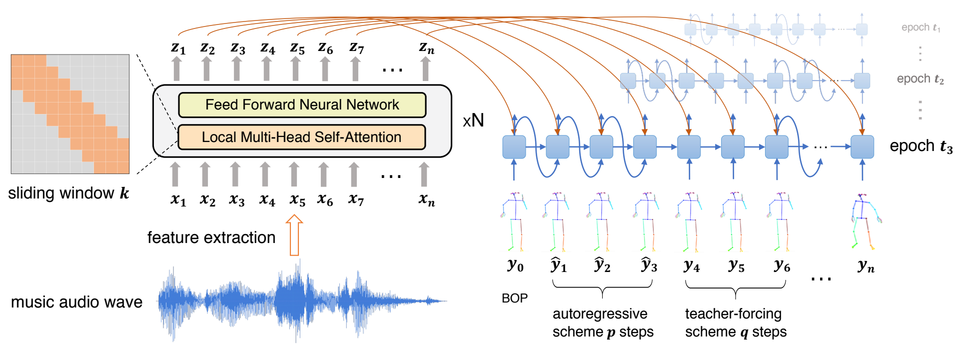

Figure 1: The overview of our method. The transformer based encoder is stacked of N identical layers with the local self-attention mechanism, which aims to efficiently process long features sequences extracted from music in a GPU-memory-economic way. The decoder exploits a recurrent structure to predict dance movement y_i frame by frame conditioned on hidden state h_i of decoder and latent vector z_i. Note that y_0 is a predefined Begin Of Pose (BOP). During the training phase, the proposed curriculum learning strategy gently increases the number of steps of autoregressive scheme according to the number of training epochs. This strategy bridges the gap between training and inference of autoregressive models, and thus alleviates error accumulation at inference.

3.1 Problem Formalization

Suppose that there is a dataset $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^N$, where $X = \{x_t\}_{t=1}^n$ is a music clip with x_t being a vector of acoustic features at time-step t, and $Y = \{y_t\}_{t=1}^n$ is a sequence of dance movements with

y_t aligned to x_t. The goal is to estimate a generation model $g(\cdot)$ from $\mathcal{D}$, and thus given a new music input X, the model can synthesize dance Y to music X based on $g(X)$. We first present our seq2seq architectures chosen for music-conditioned dance generation in the following section. Later in the experiments, we empirically justify the choice by comparing the architectures with other alternatives.

3.2 Model Architecture

In the architecture of $g(\cdot)$, a music encoder first transforms $X = (x_1, ..., x_n)$ ($x_i \in \mathbb{R}^{d_x}$) into a hidden sequence $Z = (z_1, ..., z_n)$ ($z_i \in \mathbb{R}^{d_z}$) using a local self-attention mechanism to reduce the memory requirement for long sequence modeling, and then a dance decoder exploits a recurrent structure to autoregressively predicts movements $Y = (y_1, ..., y_n)$ conditioned on Z.

Music Encoder. Encouraged by the compelling performance on music generation (Huang et al., 2018), we define the music encoder with a transformer encoder structure. While the self-attention mechanism (Vaswani et al., 2017) in transformer can effectively represent the multi-scale structure of music, the quadratic memory complexity $\mathcal{O}(n^2)$ about sequence length n impedes its application to long sequence modeling due to huge of GPU memory consumption (Child et al., 2019). To keep the effectiveness of representation and control the cost, we introduce a local self-attention mechanism that modifies the receptive field of self-attention by restricting the element connections within k-nearest neighbors. Thus, the memory complexity is reduced to $\mathcal{O}(nk)$. k could be small in our scenario since we only pursue an effective representation for a given music clip. Therefore, the local patterns of music are encoded in z_t that is sufficient for the generation of movement y_t at time-step t, which is aligned with the common sense that the dance movement at certain time-step is highly influenced by the nearby clips of music. Yet we can handle long sequences of acoustic features, e.g., more than 1000 elements, in an efficient and memory-economic way.

Specifically, we first embed $X = (x_1, ..., x_n)$ into $U = (u_1, ..., u_n)$ with a linear layer parameterized by $W^E \in \mathbb{R}^{d_x \times d_z}$. Then $\forall i \in \{1, ..., n\}$, z_i can be formulated as:

$$z_i = \mathcal{F}(a_i), \quad a_i = \sum_{j=i-\lfloor k/2 \rfloor}^{i+\lfloor k/2 \rfloor} \alpha_{ij}(u_j W_l^V), \quad U = XW^E, \tag{1}$$

where $\mathcal{F}(\cdot) : \mathbb{R}^{d_v} \to \mathbb{R}^{d_z}$ is a feed forward neural network. Each u_j is only allowed to attend its k-nearest neighbors including itself, where k is a hyper-parameter referring to the sliding window size of the local self-attention. Then attention weight α_{ij} is calculated using a softmax function as:

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{t=j-\lfloor k/2 \rfloor}^{j+\lfloor k/2 \rfloor} \exp e_{it}}, \quad e_{ij} = \frac{(u_i W_l^Q)(u_j W_l^K)^\top}{\sqrt{d_k}}, \tag{2}$$

where for the the l-th head, W_l^Q, $W_l^K \in \mathbb{R}^{d_z \times d_k}$ and $W_l^V \in \mathbb{R}^{d_z \times d_v}$ are parameters that transform U into a *query*, a *key*, and a *value* respectively. d_z is the dimension of hidden state z_i while d_k is the dimension of *query*, *key* and d_v is the dimension of *value*.

Dance Decoder. We choose a recurrent neural network as the dance decoder in consideration of two factors: (1) the chain structure can well capture the spatial-temporal dependency among human movement dynamics, which has proven to be highly effective in the state-of-the-art methods for human motion prediction (Li et al., 2018; Mao et al., 2019); (2) our proposed learning strategy is tailored for the autoregressive models like RNN, as will be described in the next section. Specifically, with $Z = (z_1, ..., z_n)$, the dance movements $Y = (\hat{y}_1, ..., \hat{y}_n)$ are synthesized by:

$$\hat{y}_i = [h_i; z_i]W^S + b, \tag{3}$$

$$h_i = \mathrm{RNN}(h_{i-1}, \hat{y}_{i-1}), \tag{4}$$

where h_i is the i-th hidden state of the decoder and h_0 is initialized by sampling from the standard normal distribution to enhance variation of the generated sequences. $[\cdot;\cdot]$ denotes the concatenation operation. $W^S \in \mathbb{R}^{(d_s+d_z)\times d_y}$ and $b \in \mathbb{R}^{d_y}$ are parameters where d_s and d_y are the dimensions of h_i and $\hat{y}_i$, respectively. At the i-th time-step, the decoder predicts the movement $\hat{y}_i$ conditioned on h_i as well as the latent feature representation z_i, and thus can capture the fine-grained correspondence between music feature sequence and dance movement sequence.

3.3 Dynamic Auto-Condition Learning Approach

Exposure bias (Ranzato et al., 2015; Lamb et al., 2016; Zhang et al., 2019) is a notorious issue in natural language generation (NLG), which refers to the train-test discrepancy of autoregressive generative models. This kind of models are never exposed to their own prediction errors at the training phase due to the conventional teacher-forcing training scheme. However, compared to NLG, motion sequence generation suffers from the more severe exposure bias problem. In NLG, the bias of predicted probability distribution over vocabulary can be corrected by sampling strategy. For instance, we can still generate the target token with index 2 by sampling on probability distribution [0.3, 0.3, 0.4] whose groundtruth is [0, 0, 1]. While any small biases of the predicted motion at each time-step will be accumulated and propagated to the future, since each generated motion is a real-valued vector (rather than a discrete token) representing 2D or 3D body keyjoints in the continuous space. As a result, the generated motion sequences tend to quickly freeze within a few seconds. This problem becomes even worse in the generation of long motion sequences, e.g., more than 1000 movements.

Scheduled sampling (Bengio et al., 2015) is proposed to alleviate exposure bias in NLG, which does not work in long motion sequence generation. Since its sampling-based strategy would feed long predicted sub-sequences of high bias into the model at early stage, which causes gradient vanishing at training due to error accumulation of these high-biased sub-sequences. Motivated by this, we propose a dynamic auto-condition training method as a novel curriculum learning strategy to alleviate the error accumulation of autoregressive models in long motion sequence generation. Specifically, our learning strategy gently changes the training process from a fully guided teacher-forcing scheme using the previous ground-truth movements, towards a less guided autoregressive scheme mostly using the generated movements instead, which bridges the gap between training and inference.

As shown in Figure 1, The decoder predicts $Y_{tgt} = \{y_1, ..., y_p, y_{p+1}, ..., y_{p+q}, y_{p+q+1}, ...\}$ from $Y_{in} = \{y_0, \hat{y_1}, ..., \hat{y_p}, y_{p+1}, ..., y_{p+q}, \hat{y}_{p+q+1}, ...\}$ which alternates the predicted sub-sequences with length p (e.g., $\hat{y}_1, ..., \hat{y}_p$) and the ground-truth sub-sequences with length q (e.g., $y_{p+1}, ..., y_{p+q}$). y_0 is a predefined begin of pose (BOP). During the training phase, we fix q and gradually increase p according to a growth function $f(t) \in \{const, \lfloor \lambda t \rfloor, \lfloor \lambda t^2 \rfloor, \lfloor \lambda e^t \rfloor\}$ where t is the number of training epochs and $\lambda < 1$ controls how frequently p is updated. We choose $f(t) = \lfloor \lambda t \rfloor$ in practice through the empirical comparison. Note that our learning strategy degenerates to the method proposed in Li et al. (2017) when $f(t) = const$. While the advantage of a dynamic strategy over a static strategy lies in that the former introduces the curriculum learning spirit to dynamically increase the difficulty of curriculum (e.g., larger p) as the model capacity improves during training, which can further improve the model capacity and alleviate the error accumulation of autoregressive predictions. Finally, we estimate parameters of $g(\cdot)$ by minimizing the ℓ_1 loss on $\mathcal{D}$:

$$\ell_1 = \frac{1}{N} \sum_{i=1}^{N} ||g(X_i) - Y_{tgt}^{(i)}||_1 \tag{5}$$

4 Experimental Setup

4.1 Dataset Collection and Preprocessing

Although the dataset (Lee et al., 2019) contains about 70-hour video clips, the average length of each clip is only about 6 seconds. Moreover, their dataset cannot be used by other methods to generate long-term dance except for their specially designed decomposition-to-composition framework. To facilitate the task of long-term dance generation with music, we collect a high-quality dataset consisting of 790 one-minute clips of pose data with 15 FPS, totaling about 12 hours. And there are three styles in the dataset including "Ballet", "Hiphop" and "Japanese Pop". Table 1 shows the statistics of our dataset. In the experiment, we randomly split the dataset into the 90% training set and 10% test set.

Pose Preprocess. For the human pose estimation, we leverage OpenPose (Cao et al., 2017) to extract 2D body keyjoints from videos. Each pose consists of 25 keyjoints[2] and is represented by a 50-dimension vector in the continuous space. In practice, we develop a linear interpolation algorithm to find the missing keyjoints from nearby frames to reduce the noise in extracted pose data.

[2]https://github.com/CMU-Perceptual-Computing-Lab/openpose/blob/master/doc/output.md#pose-output-format-body_25

Table 1: Statistics of the dataset.

Category	Num of Clips	Clip Length (min)	FPS	Resolution
Ballet	136	1	15	720P
Hiphop	298	1	15	720P
Japanese Pop	356	1	15	720P

Audio Preprocess. Librosa (McFee et al., 2015) is a well-known audio and music analysis library in the music information retrieval, which provides flexible ways to extract the spectral and rhythm features of audio data. Specifically, we extract the following features: *mel frequency cepstral coefficients (MFCC)*, *MFCC delta*, *constant-Q chromagram*, *tempogram* and *onset strength*. To better capture the beat information of music, we convert *onset strength* into a one-hot vector as an additional feature to explicitly represent the beat sequence of music, i.e., *beat one-hot*. Finally, we concatenate these audio features as the representation of a music frame, as shown in Table 4 in Section 6.2. During extracting audio features, the sampling rate is 15,400Hz and hop size is 1024. Hence, we have 15 audio samples per second that is aligned with the 15 FPS of pose data.

4.2 Implementation Details

The music encoder consists of a stack of $N = 2$ identical layers. Each layer has two sublayers: a local self-attention sublayer with $l = 8$ heads and a position-wise fully connected feed-forward sublayer with 1024 hidden units. Each head contains a scaled dot-product attention layer with $d_k = d_v = 64$ and its receptive yield is restricted by setting $k = 100$. Then we set the sequence length $n = 900$, dimension of acoustic feature vector $d_x = 438$, dimension of hidden vector $d_z = 256$, respectively. The dance decoder is a 3-layer LSTM with $d_s = 1024$ and the dimension of pose vector $d_y = 50$. We set $\lambda = 0.01$ and $q = 10$ for the proposed learning approach and train the model using the Adam optimizer with the learning rate 1e-4 on 2 NVIDIA V100 GPUs.

4.3 Baselines

The music-conditioned generation is an emerging task and there are few methods proposed to solve this problem. In our experiment, we compare our proposed approach to the following baselines: (1) **Dancing2Music.** We use Dancing2Music (Lee et al., 2019) as the primary baseline that is the previous state-of-the-art; (2) **Aud-MoCoGAN.** Aud-MoCoGAN is an auxiliary baseline used in Dancing2Music, which is original from MoCoGAN (Tulyakov et al., 2018); (3) **LSTM.** Shlizerman et al. (2018) propose a LSTM network to predict body dynamics from the audio signal. We modify their open-source code to take audio features of music as inputs and produce human pose sequences.

5 Experimental Results

In this section, we conduct extensive experiments to evaluate our approach and compare it to the aforementioned baselines. Note that we do not include comparison experiment on the dataset collect by Lee et al. (2019) due to the above issue in Section 4.1.

5.1 Automatic Metrics and Human Evaluation

We evaluate the different methods by automatic metrics and conduct a human evaluation on motion realism and smoothness of generated dance movements, as well as style consistency. Specifically, we randomly sample 60 music clips from the test set to generate dances, then divide the dances generated by three methods and the real dances into four pairs: (LSTM, ours), (Dancing2Music, ours), (real, ours) and (real, real). We invite 10 amateur dancers as the annotators to answer 3 questions for each pair that is blind for them (use (A, B) pair instead): (1) *Which dance is more realistic regardless of music?* (2) *Which dance is more smooth regardless of music?* (3) *Which dance matches the music better with respect to style?* Note that, we do not include Aud-MoCoGAN in the human evaluation, since both it and Dancing2Music are GAN based methods while the latter is the state-of-the-art.

Table 2: **Automatic metrics of different methods**. *FID* (lower is better) evaluates the quality and realism of dances by measuring the distance between the distributions of the real dances and the generated dances. *ACC* evaluates the style consistency between generated dances and music. *Beat Coverage* measures the ratio of total kinematic beats to total musical beats. The higher the beat coverage is, the stronger the rhythm of dance is. *Beat Hit Rate* measures the ratio of kinematic beats aligned with musical beats to total kinematic beats. *Diversity* denotes the variations among a set of generated dances while *Multimodality* refers to the variations of generated dances for the same music.

Method	FID	ACC (%)	Beat Coverage (%)	Beat Hit Rate (%)	Diversity	Multimodality
Real Dances	2.6	99.5	56.4	63.9	40.2	-
LSTM	51.9	12.1	4.3	9.7	16.8	-
Aud-MoCoGAN	48.5	33.8	10.9	28.5	30.7	16.4
Dancing2Music	22.7	60.4	15.7	65.7	30.8	**18.9**
Ours	**6.5**	**77.6**	**21.8**	**68.4**	**36.9**	15.3

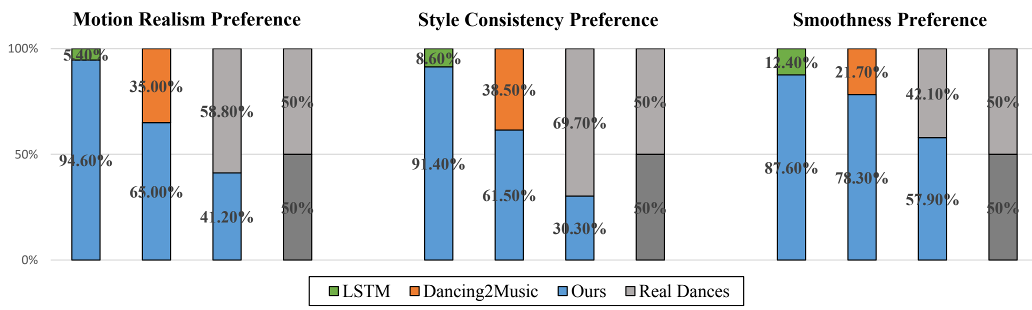

Figure 2: **Human evaluation results on motion realism, style consistency and smoothness.** We conduct a human evaluation to ask annotators to select the dances that are *more realistic and more smooth regardless of music, more style-consistent with music* through pairwise comparison.

Motion Realism, Style Consistency and Smoothness. We evaluate the visual quality and realism of generated dances by Fréchet Inception Distance (FID) (Heusel et al., 2017), which is used to measure how close the distribution of generated dances is to the real. Similar to Lee et al. (2019), we train a style classifier on dance movement sequences of three styles and use it to extract features for the given dances. Then we calculate FID between the synthesized dances and the real. As shown in Table 2, our FID score is significantly lower than those of baselines and much closer to that of the real, which means our generated dances are more motion-realistic and more likely to the real dances. Besides, we use the same classifier to measure the style accuracy of generated dance to the music, and our approach achieves 77.6% accuracy and significantly outperforms Dancing2Music by 17.2%.

Human evaluation in Figure 2 consistently shows the superior performance of our approach, compared to baselines on motion realism, style consistency and smoothness. We observe the dances generated by LSTM have lots of floating movements and tend to quickly freeze within several seconds due to the error accumulation, which results in lower preferences. While Dancing2Music can generate the smooth dance units, the synthesized dances still have significant jumps where dance units are connected due to the inherent gap between them, which harm its overall performance in practice. This is also reflected in the preference comparisons to our approach, in which only 35% annotators prefer Dancing2Music on motion realism and 21.7% prefer Dancing2Music on smoothness. Besides, the result on style consistency shows our approach can generate more style-consistent dances with musics. Since our approach models the fine-grained correspondence between music and dance at the methodological level and also introduces more specific music features. However, Dancing2Music only considers a global style feature extracted by a pre-trained classifier in the dance generation.

In the comparison to real dances, 41.2% annotators prefer our method on motion realism while 30.3% prefer on style consistency. Additionally, we found that 57.9% annotators prefer our approach on smoothness compared to real dances. Since the imperfect OpenPose (Cao et al., 2017) introduces the minor noise on pose data extracted from dance videos. On the other hand, in the preprocessing, we develop an interpolation algorithm to find missing keyjoints and reduce the noise of training pose data.

Besides, this also indicates the chain structure based decoder can well capture the spatial-temporal dependency among human motion dynamics and produce the smooth movements.

Beat Coverage and Hit Rate. We also evaluate the beat coverage and hit rate introduced in Lee et al. (2019), which defines the beat coverage as B_k/B_m and beat hit rate as B_a/B_k, where B_k is the number of kinematic beats, B_m is the number of musical beats and B_a is the number of kinematic beats that are aligned with the musical beats. An early study (Ho et al., 2013) reveals that kinematic beat frames occur when the direction of the movement changes. Therefore, similar to Yalta et al. (2019), we use the standard deviation (SD) of movement to detect the kinematic beats in our experiment. The onset strength (Ellis, 2007) is a common way to detect musical beats. Figure 3 shows two short clips of motion SD curves and aligned musical beats. We observed that the kinematic beats occur where the musical beats occur, which is consistent with the common sense that dancers would step on musical beats during dancing but do not step on every musical beat. As we can see in Table 2, LSTM and Aud-MoCoGAN generate dances with few kinematic beats and most of them do not match the musical beats. Our method outperforms Dancing2Music by 6.1% on beat coverage and 2.7% on hit rate, which indicates that introducing the features about musical beat into model is beneficial to the better beat alignment of generated dance and input music.

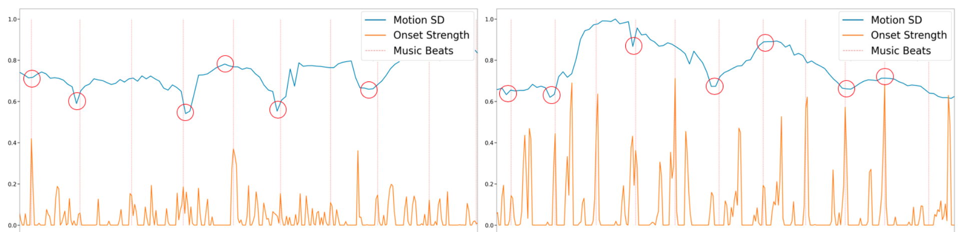

Figure 3: Beat tracking curves for music and dance by onset strength and motion standard deviation respectively. The left is a short clip of ballet while the right is a short clip of hip-hop. The red circles are kinematic beats and dash lines denote the musical beats.

Diversity and Multimodality. Lee et al. (2019) introduce the diversity metric and the multimodality metric. The diversity metric evaluates the variations among generated dances corresponding to various music, which reflects the generalization ability of the model and its dependency on the input music. Similarly, we use the average feature distance as the measurement and these features are extracted by the same classifier used in measuring FID. We randomly sample 60 music clips from test set to generate dances and randomly pick 500 combinations to compute the average feature distance. Results in Table 2 show the performance of our approach is superior to Dancing2Music on diversity. The reason is that Dancing2Music only considers a global music style feature in generation while different music of the same style have the almost same style feature. Our approach models the fine-grained correspondence between music and dance at the methodological level, simultaneously takes more specific music features into account. In other words, our approach is more dependent on input music and thus can generate different samples for difference music clips of the same style.

Multimodality metric evaluates the variations of generated dances for the same music clip. Specifically, we generate 5 dances for each of randomly sampled 60 music clips and compute the average feature distance. As we can see, our approach slightly underperforms Dancing2Music and Aud-MoCoGAN since it has more dependency on music inputs as aforementioned, which make the generated dances have the limited variation patterns given the same music clip.

5.2 Long-Term Dance Generation

To further investigate the performance of different methods on long-term dance generation, we evaluate FID of dances generated by different methods over time. Specifically, we first split a generated dance with 1 minute into 15 four-second clips and measure FID of each clip. Figure 4 shows FID scores of LSTM and Aud-MoCoGAN grow rapidly in the early stage due to error accumulation and converge to the high FID, since the generated dances quickly become frozen. Dancing2Music maintains the relatively stable FID scores all the time, which benefits from its decomposition-to-composition method. While its curve still has subtle fluctuations since it synthesizes whole dances by

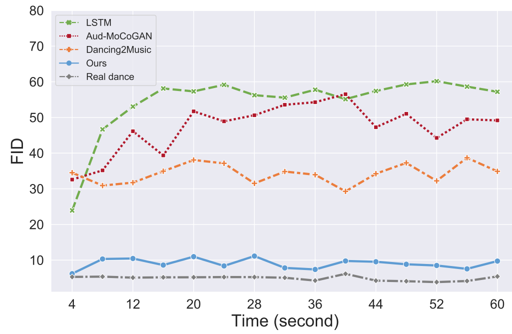

Figure 4: FID curves of different methods over time.

Table 3: Left: Automatic metrics of our model with different encoder structures. Right: Automatic metrics of our model training by different learning strategies.

Encoder Structure	FID	ACC (%)
LSTM	7.9	45
ConvS2S encoder	13.3	54
Global self-attention	3.6	80
Local self-attention	4.3	78.5

Learning Approach	FID	ACC (%)
Teacher-forcing	61.2	5.3
Auto-condition ($const$)	15.7	35
Ours ($\lfloor \lambda e^t \rfloor$)	9.8	69
Ours ($\lfloor \lambda t^2 \rfloor$)	6.4	73
Ours ($\lfloor \lambda t \rfloor$)	5.1	77.6

composing the generated dance units. Compared to baselines, FID of our method are much lower and close to real dances, which validates the good performance of our method on long-term generation.

5.3 Ablation Study

Comparison on Encoder Structures. We compare the different encoder structures in our model with the same LSTM decoder and the same curriculum learning strategy, as shown in Table 3. The transformer encoders outperform LSTM and the encoder in ConvS2S (Gehring et al., 2017) on both FID and style consistency, due to its superior performance on sequence modeling. Although the transformer encoder with our proposed local self-attention slightly underperforms the global self-attention by 0.7 higher in FID and 1.5% lower in ACC, it has much fewer parameters in our settings. This confirms the effectiveness of the local self-attention on long sequence modeling.

Comparison on Learning Strategies. To investigate the performance of our proposed learning approach, we conduct an ablation study to compare the performances of our model using different learning strategies. As we can see the right of in Table 3, all curriculum learning strategies significantly outperform the static auto-condition strategy (Li et al., 2017). Among them, the linear growth function $f(t) = \lfloor \lambda t \rfloor$ performs best, which might be that increasing the difficulty of curriculum too fast would result in the model degradation. Besides, the original teacher-forcing strategy has the highest FID and the lowest style accuracy due to severe error accumulation problem, which indicates that using our proposed curriculum learning strategy to train the model can effectively alleviate this issue.

6 Conclusion

In this work, we propose a novel seq2seq architecture for long-term dance generation with music, e.g., about one-minute length. To efficiently process long sequences of music features, we introduce a local self-attention mechanism in the transformer encoder structure to reduce the quadratic memory requirements to the linear in the sequence length. Besides, we also propose a dynamic auto-condition training method as a novel curriculum learning strategy to alleviate the error accumulation of autoregressive models in long motion sequence generation, thus facilitate the long-term dance generation. Extensive experiments have demonstrated the superior performance of our proposed approach on automatic metrics and human evaluation. Future works include the exploration on end-to-end methods that could work with raw audio data instead of preprocessed audio features.

REFERENCES

Janet Adshead-Lansdale and June Layson. *Dance history: An introduction*. Routledge, 2006.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pp. 1171–1179, 2015.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7291–7299, 2017.

Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5933–5942, 2019.

Joyce L Chen, Virginia B Penhune, and Robert J Zatorre. Listening to musical rhythms recruits motor regions of the brain. *Cerebral cortex*, 18(12):2844–2854, 2008.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2262–2271, 2019.

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

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

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

Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4346–4354, 2015.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1243–1252. JMLR. org, 2017.

Partha Ghosh, Jie Song, Emre Aksan, and Otmar Hilliges. Learning human motion models for long-term predictions. In *2017 International Conference on 3D Vision (3DV)*, pp. 458–466. IEEE, 2017.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pp. 6626–6637, 2017.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

Chieh Ho, Wei-Tze Tsai, Keng-Sheng Lin, and Homer H Chen. Extraction and alignment evaluation of motion beats for street dance. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2429–2433. IEEE, 2013.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. *arXiv preprint arXiv:1809.04281*, 2018.

Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 5308–5317, 2016.

Alex M Lamb, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pp. 4601–4609, 2016.

Hsin-Ying Lee, Xiaodong Yang, Ming-Yu Liu, Ting-Chun Wang, Yu-Ding Lu, Ming-Hsuan Yang, and Jan Kautz. Dancing to music. In *Advances in Neural Information Processing Systems*, pp. 3581–3591, 2019.

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

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

Andreas M Lehrmann, Peter V Gehler, and Sebastian Nowozin. Efficient nonlinear markov models for human motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1314–1321, 2014.

Chen Li, Zhen Zhang, Wee Sun Lee, and Gim Hee Lee. Convolutional sequence to sequence model for human dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5226–5234, 2018.

Zimo Li, Yi Zhou, Shuangjiu Xiao, Chong He, Zeng Huang, and Hao Li. Auto-conditioned recurrent networks for extended complex human motion synthesis. *arXiv preprint arXiv:1707.05363*, 2017.

Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 375–383, 2017.

Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9489–9497, 2019.

Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 2015.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

Xuanchi Ren, Haoran Li, Zijian Huang, and Qifeng Chen. Music-oriented dance video synthesis with pose perceptual loss. *arXiv preprint arXiv:1912.06606*, 2019.

Eli Shlizerman, Lucio Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman. Audio to body dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7574–7583, 2018.

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

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

Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis. Modeling human motion using binary latent variables. In *Advances in neural information processing systems*, pp. 1345–1352, 2007.

Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1526–1535, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Jack Wang, Aaron Hertzmann, and David J Fleet. Gaussian process dynamical models. In *Advances in neural information processing systems*, pp. 1441–1448, 2006.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *arXiv preprint arXiv:1808.06601*, 2018.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057, 2015.

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

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

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 5907–5915, 2017.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. *arXiv preprint arXiv:1906.02448*, 2019.

APPENDIX

6.1 MORE DETAILS ABOUT DATA COLLECTION

The dance videos are collected from YouTube by crowd workers, which are all solo dance videos with 30FPS. We trim the beginning and ending several seconds for each of collected videos to remove the silence parts, and split them into one-minute video clips. Then, we extract 2D pose data from these video clips by OpenPose (Cao et al., 2017) with 15FPS setting and collect 790 one-minute clips of pose data with 15FPS, totaling about 13 hours. Finally, we extract the corresponding audio data from video clips by FFmpeg[3]. The code is implemented based on PyTorch framework. MIT License will be used for the released code and data.

6.2 EXTRACTED MUSIC FEATURES

In this section, we detail the extracted features of music that are feeded into our model. Specifically, we leverage the public Librosa (McFee et al., 2015) to extract the music features including: 20-dim *MFCC*, 20-dim *MFCC delta*, 12-dim *chroma*, 384-dim *tempogram*, 1-dim *onset strength* (i.e., *envelope*) and 1-dim *one-hot beat*, as shown in Table 4.

Table 4: Acoustic features of music.

Feature	Characteristic	Dimension
MFCC	Pitch	20
MFCC delta	Pitch	20
Constant-Q chromagram	Pitch	12
Tempogram	Strength	384
Onset strength	Strength	1
Beat one-hot	Beat	1

6.3 DOWNSTREAM APPLICATIONS

The downstream applications of our proposed audio-conditioned dance generation method include: (1) Our model can be used to help professional people choreograph new dances for a given song and teach human how to dance with regard to this song; (2) With the help of 3D human pose estimation (Ci et al., 2019) and 3D animation driving techniques, our model can be used to drive the various 3D character models, such as the 3D model of Hatsune Miku (very popular virtual character in Japan). This technique has the great potential for the virtual advertisement video generation that could be used for the promotion events on social medias like TikTok, Twitter and etc.

6.4 MUSICAL BEAT DETECTION UNDER LOW SAMPLING RATES

In this section, we conduct an additional experiment to evaluate the performance of Librosa beat tracker under 15,400Hz sampling rate. Specifically, we first utilize it to extract the onset beats from audio data with 15,400Hz and 22,050Hz respectively, then compare two groups of onset beats under the same time scale. Since 22,050Hz is a common sampling rate used in music information retrieval, we define the beat alignment ratio as B_2/B_1, where B_1 is the number of beats under 15,400Hz and B_2 is the number of beats under 15,400Hz that are aligned with the beats under 22,050Hz. One beat is counted when $\mid t_1 - t_2 \mid \leq \Delta t$. t_1 denotes the timestamp of a certain beat under 15,400Hz while t_2 refers to the timestamp of a certain beat under 22,050Hz, Δt is the time offset threshold.

In the experiment, we set $\Delta t = 1/15s$ (FPS is 15) and calculate the beat alignment ratio for audio data of three styles respectively. We randomly sample 10 audio clips from each style to calculate the beat alignment ratio and do the sampling for 10 times to take the average. As shown in Table 5, most of beats extracted with 15,400Hz are aligned with those extracted with 22,050Hz. Besides, we randomly sample an audio clip and visualize beat tracking curves of its first 20 seconds under the sampling rates of 15,400Hz and 22,050Hz. As we can see in Figure 5, most of the beats from 15,400Hz and 22,050Hz are aligned within the time offset threshold $\Delta t = 1/15s$.

[3]`https://ffmpeg.org/`

Table 5: The beat alignment ratio B_2/B_1.

Category	Beat Alignment Ratio (%)
Ballet	88.7
Hiphop	93.2
Japanese Pop	91.6

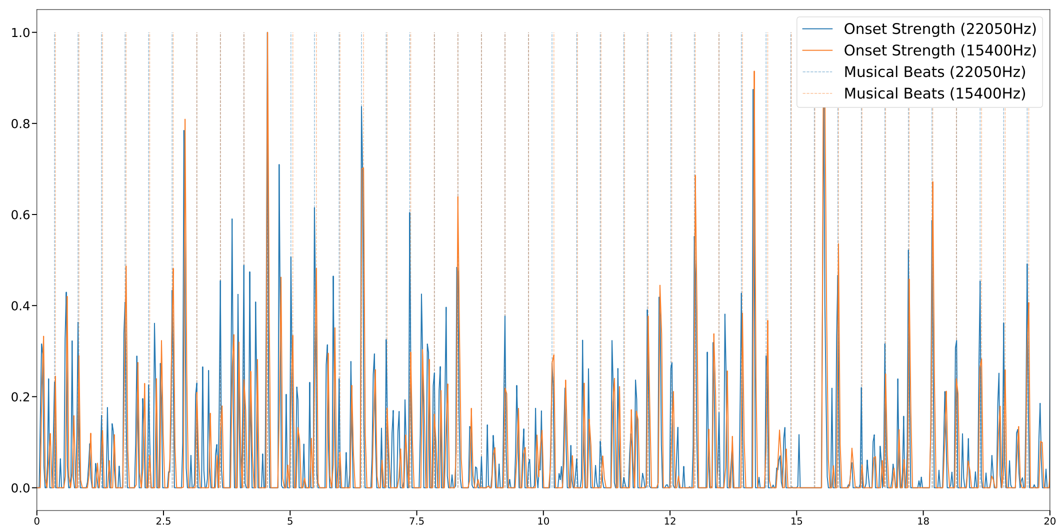

Figure 5: Beat tracking curves for the first 20 seconds of a music clip randomly sampled from audio data under the sampling rates of 15,400Hz and 22,050Hz. Most of the beats are aligned within the time offset threshold $\Delta t = 1/15s$.