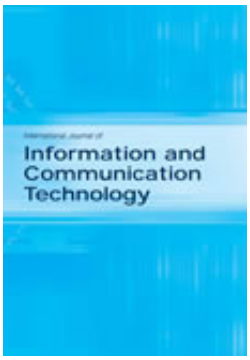


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Intelligent generation of robotic dance motions via convolution-enhanced transformer networks

Fei Yue, Jing Tong, Zhen Ren

DOI: 10.1504/IJICT.2025.10072511

Article History:

Received:	02 June 2025
Last revised:	16 June 2025
Accepted:	16 June 2025
Published online:	30 July 2025

Intelligent generation of robotic dance motions via convolution-enhanced transformer networks

Fei Yue*

Hubei Engineering University,
Xiaogan 432000, China
Email: yuefei0428@163.com
*Corresponding author

Jing Tong

Hubei University of Automotive Technology,
Shiyan 442000, China
Email: tongjing202502@163.com

Zhen Ren

Wuhan Conservatory of Music,
Wuhan 430000, China
Email: rZ1375915023@163.com

Abstract: The generation of robotic dance movements represents a complex and challenging task. To tackle the limitations of current research, such as poor coherence in movement sequences and low generation efficiency, this paper proposes a depth-separable convolution-enhanced Transformer network (DSFormer). DSFormer significantly reduces the number of parameters while enhancing the computational efficiency of the model. Furthermore, based on DSFormer, a music encoder, a robot dance movement encoder, and a cross-modal generator are developed. These components effectively capture both local spatial features and global temporal characteristics of music and robotic dance motion sequences, thereby alleviating the adverse effects of noisy data. Experimental comparisons conducted on real-world datasets reveal that the proposed method achieves at least a 21.64% reduction in the Frechet Inception Distance (FID) score compared to baseline approaches. This not only ensures the generation of high-quality dance motions but also maintains precise synchronisation with the music.

Keywords: dance action generation; depth separable convolution; transformer model; music encoder; cross-modal generator.

Reference to this paper should be made as follows: Yue, F., Tong, J. and Ren, Z. (2025) 'Intelligent generation of robotic dance motions via convolution-enhanced transformer networks', *Int. J. Information and Communication Technology*, Vol. 26, No. 30, pp.97–112.

Biographical notes: Fei Yue received her PhD from University of Perpetual Help System-DALTA (UPHSD) in 2023. Since 2010, she has been teaching at the Hubei Engineering University. Her research interests include digital intelligence and communication of dance.

Jing Tong received her PhD from University of Perpetual Help System-DALTA (UPHSD) in 2023. Since 2023, she has been teaching at the Hubei University of Automotive Technology. Her research interests include digital media and space display design.

Zhen Ren received her technical secondary school education from Wuhan Conservatory of Music Affiliated Middle School in 2019, and received her Bachelor's from Wuhan Conservatory of Music in 2023. Her research interests include digital intelligence and dance communication.

1 Introduction

Dance, as an ancient and expressive art form, carries human culture, emotion and creativity (Liu and Hu, 2021). Combining the art of dance with robotics to endow robots with the ability to perform dance has become a highly innovative and forward-looking research direction (Li et al., 2020). Robot dance can not only bring a new visual experience to the audience, but also play an important role in entertainment, education, and cultural dissemination. However, it is not easy to realise high-quality intelligent generation of robotic dance movements and faces many challenges (Qin et al., 2018). Dance movements are highly complex and diverse. Robots need to be able to accurately capture these features and generate smooth and natural dance movement sequences, which requires algorithms with strong pattern recognition and movement generation capabilities (Peng et al., 2021). Traditional research generates dance movements based on predefined dance rules and movement templates, which is simple and direct, but less flexible and difficult to generate diverse and creative dance movements (Peng et al., 2022). Thus, it is highly valuable in practice to explore an efficient method for intelligent generation of robot dance movements.

In earlier work, researchers solved the dance movement generation task as a matching problem, the basic idea of this method is to model the task of generating dance movements as a matching problem between the input (such as music, text description) and the movement sequence. By seeking the best matching relationship between the input and the movements, generate the dance movements that meet the requirements. And select or generate the most suitable action sequence based on the matching degree. In the problem of action matching, the optimal path refers to the best mapping path from the input to the action sequence, which makes the matching degree the highest or the loss the least. Valle-Pérez et al. (2021) to a one-to-many or many-to-many mapping of dances. Lee and Lee (2019) used a genetic algorithm to learn the correlation among music and dance, and then constructed a motion graph model to produce the corresponding dance segment in light of the music, but the quality of the generation was not high. The dance motions produced through the above approaches are often incoherent and unnatural. To ensure the coherence between candidate movements, Wang et al. (2024) utilised transition frame interpolation to solve the problem of incoherent retrieved movements. Xu et al. (2023) set different base movements for the robot and used the extracted music feature parameters to retrieve the corresponding body movements.

Deep learning automatically learns spatio-temporal patterns in action sequences through DNN, avoiding the tedious manual feature design process of traditional methods. Ahn et al. (2020) proposed an autoregressive codec network using spectrograms with

dance movement sequences as input to train the autoregressive codec network to generate new dance movements. Kritsis et al. (2022) improved the generation by processing audio features through convolutional neural network (CNN), splicing real dance movement sequences with audio features and inputting them into long short-term memory network (LSTM) to decode them into future dance movement sequences. Liu and Ko (2022) proposed a dance generation model based on time-convolutional LSTM, which uses time-convolutional LSTM to generate dance movements, in addition to introducing dance melody lines to improve controllability. Zhuang et al. (2022) also proposed an autoregressive generation model to generate 3D dance movements using the style, rhythm and melody of the music as control signals. The multi-dimensional features of music are integrated through multimodal technology, which improves the generation quality of action images.

Generative adversarial networks (GANs) show unique advantages in the task of robotic dance movement generation, and their core value lies in addressing the inherent shortcomings of traditional generative models through an adversarial training mechanism. Han et al. (2024) suggested a GAN-based framework for robotic dance movement generation, where the generator is composed of an audio encoder, a bi-directional gate rate unit (GRU), and a pose generator. The discriminator is constituted by a global content discriminator as well as a local timing discriminator, which improves the clarity of image generation. Liang et al. (2024) proposed a GAN-based framework to generate a model of robotic dance movements from music by seamlessly splicing multiple basic dance units based on input music. GAN-based action generation methods are difficult to cope with long time-series modelling, while Transformer can explicitly capture the relationship between joint points that are far apart in dance movements, which significantly improves the effect of action generation. Zhou et al. (2024) used Transformer to learn the features of music and movement separately and designed a cross-modal Transformer to generate dance movements with good generation quality. Sun and Wang (2024) based Transformer's encoder which uses a local self-attention mechanism to process long feature sequences extracted from music. And the decoder utilises LSTM to generate the dance motion frame by frame and achieved good generation results.

Taking the above analysis into account, it is clear that the existing studies not only need to generate long time continuous movements with high complexity, but also need to capture the nonlinear relationship between movements and actions, and generate dance movements matching the music, so the study of robotic dance movement generation is a very challenging task. Local and global dependencies are present in music sequences as well as dance movement sequences, and existing studies do not sufficiently consider the coherence of movements, resulting in poor generation efficiency. Some studies may pay more attention to the fluency of the action, the matching degree with the input (such as music), or the physical rationality, rather than directly emphasising the nonlinear relationship. For this reason, this paper offers an approach for intelligent generation of robot dance movements based on convolutionally enhanced transformer. The chief work of the approach is summarised as bellow.

1 In response to the large number of parameters in transformer's multi-headed self-attention module, this paper firstly takes advantage of the linear complexity of pooling and sequence length as well as the advantage of being parameter-free, and proposes DSFormer. Introduces a parallel pooling module instead of the

multi-headed self-attention module, and replaces the conventional convolution with a depth-separable convolution to further reduce the amount of parameters.

2 Construction of a music encoder, a robot dance movement encoder, and a cross-modal data generator based on DSFormer. DSFormer identifies the local spatial characteristics as well as the global characteristics within the time- series of music and robot dance sequences. Consequently, it detects the local and global dependencies among the sequences and alleviates the influence of noisy information.

3 A global content discriminator is used to seize the global correlation among music and dance. The robot dance sequence that has been generated is partitioned into a plurality of overlapping subsequences. Following this, these subsequences are delivered to a motion discriminator, which is a two-branch convolutional network. Finally, the output of the motion discriminator is used to determine the authenticity of the subsequences using a classifier that outputs different scores.

4 Extensive comparative experiments were conducted with the benchmark method on a publicly available dataset, and the proposed method outperforms the benchmark method in all evaluation metrics, not only by producing long and coherent dance movements, but also by generating dance movements that match the music.

2 Relevant technologies

2.1 Convolutional neural network

CNN is one of the most representative deep neural networks (DNN) in deep learning, which is a deep feed-forward neural network with representation learning capability (Vonder Haar et al., 2023). Compared to traditional DNNs, the core features of CNNs originate from the simulation of biological visual cortex mechanisms, which significantly reduces the model complexity and improves the feature representation. Through designs such as local perception, weight sharing and hierarchical extraction of features, CNN has the following advantages over traditional fully connected networks in data tasks with spatial structures such as images and videos. First of all, the parameter efficiency is high. Local perception and weight sharing significantly reduce the number of parameters and lower the computational complexity. Secondly, it has strong generalisation ability. Weight sharing and hierarchical feature extraction enable the model to learn more universal features and enhance the generalisation ability. Finally, the feature representation ability is strong. Hierarchical extraction enables the model to learn abstract features from low to high levels, making it suitable for complex visual tasks. CNN has the similar structure of input level, obscured level and output layer as DNN (Liu et al., 2016), in which the hidden level contains convolutional level, pooling level and fully linked level.

1 Convolutional level. This level is used for feature extraction. While processing the image each convolutional kernel processes only one domain of the input picture, this is because the picture input is a high dimensional input, while processing each convolutional kernel in the network is fully connected to the previous layer will generate a huge amount of computation, and it is not possible to complete the

training. Therefore, each convolutional kernel is connected to only one region of the input image (Zhou, 2020).

2 Pooling level. The idea of this level is derived from the visual system of living organisms, and its role is to perform downsampling operations to downsize the feature map. The process of compressing the feature information extracted from the convolutional layer and using higher-level characteristics to symbolise the input is often called downsampling. The pooling level can efficiently decrease the information redundancy and prevent overfitting (Cong and Zhou, 2023).

3 Fully linked level. This level expands the feature map into vectors and inputs the activation function, which is used to accomplish the learning objective.

As the amount of levels rises, gradient vanishing results in a decrease in the dissimilarity between features captured by CNNs. ResNet effectively solves the degree vanishing problem of traditional CNNs in deep network training by introducing residual learning and jump connections. Formally, the process of residual unit is implied in equation (1), where $F(x)$ is the learning objective, $H(x)$ is the output, and x is the input part of the model. In ResNet, each residual unit passes the input directly to the output through a shortcut or skip operation, a phase called residuals.

$$H(x) = F(x) + x \tag{1}$$

2.2 Transformer model

The transformer model has shown significant advantages in action generation tasks due to its unique self-attention mechanism and global modelling capability (Ma et al., 2019). RNNs are commonly used to deal with long sequence generation problems, but when two time steps are too far apart, recurrent neural networks (RNNs) are unable to capture their relationship. Variants such as LSTM and GRU have been proposed to alleviate the problems in RNNs, but these models are limited to sequence context processing and the ability to model long distance dependencies. In contrast, the Transformer does not require any RNN framework, which allows it to compute in parallel, greatly improving the efficiency of training and inference.

Transformer mainly contains patch module, location coding, multi-head self-attention mechanism and feed-forward neural network level (Friedman et al., 2023). he input data undergoes processing via a multi-head self-attention level followed by a location – aware feed – forward neural network level, and residual linkage and level standardisation are used to optimise the model performance. The transformer adopts an invariable implicit vector size D in each levels, consequently, the image blocks are transformed into a one – dimensional form and then projected onto a D-dimensional space through a trainable linear projection, the output of which is called patch embedding.

In the transformer model, a typical approach to position encoding is to use the different frequencies of the sine and cosine functions as position encoding. Specifically, each row of the position encoding matrix related to a position and all columns related to a dimension. The positional encoding is computed as shown in equation (2), where w is the corner frequency, k is the integer index, ω_k is a value based on the positional information and dimension, i is the dimension, d is the length of the vector, and pos is the position of all elements in the sequence.

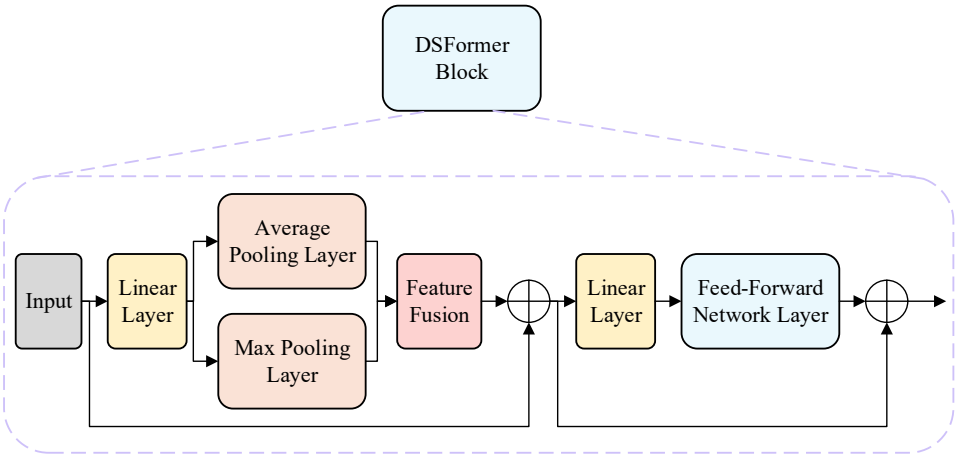
$$PE_{(pos,i)} = \begin{cases} cos\left(pos \cdot \omega_k\right) & if \ i = 2k+1 \\ sin\left(pos \cdot \omega_k\right) & if \ i = 2k \end{cases} \tag{2}$$

3 Deeply separable convolutionally enhanced transformer model

3.1 Overview of the DSFormer model

The intelligent generation model of robot dance movements usually involves a large amount of image data and complex model computation. Optimising the computational efficiency of the model can save computational resources, reduce energy consumption, and reduce the requirements for hardware equipment, making the generation of robot dance movements more feasible. By decreasing the amount of parameters in the Transformer, the calculation resources required for model operation can be reduced, thus accelerating the inference process and improving the real-time performance and responsiveness of the model. For this reason, this chapter proposes the DSFormer network, which will be introduced in the following.

Figure 1 The structure of the DSFormer module (see online version for colours)



The size of the input picture of the DSFormer is $H \times W \times C$, where H and W denote the width and height of the input picture and C denotes the amount of channels of the image. The whole network structure contains four stages, the first phase includes the patch embedding module and the DSFormer module, and the remaining three stages consist of the deep separable convolution (DSPE) module and the DSFormer block. In DSFormer network, the input image is first entered into patch embedding to perform convolutional operation through which the image is divided into blocks. After that the image is converted into a sequence into DSFormer blocks for feature extraction. After four stages of processing, the final classification result is obtained through MLP head.

The DSFormer module consists of two residual sub-modules, as shown in Figure 1. The first sub-module is composed of a linear level and a parallel pooling level, where the linear layer combines and downscales the input features to better fit the processing of the

subsequent modules. The parallel pooling level replaces the multi-head self-attention module as a mixer for the image block sequence, and extracts the global context information and local information through the parallel average pooling level and maximum pooling level. The fused features are then added to the original input through residual concatenation to perform feature enhancement, and the resulting features are input to the next residual block. The second residual submodule contains a linear layer and a feed-forward network (FFN) level for dimensional transformation, where the linear layer upscales the input features and adjusts them to a dimension that the FFN layer can handle. The FFN level effectively performs nonlinear transformation and feature extraction on the output of the linear level.

3.2 Deeply separable convolutional replacement patch embedding module

The DSPE module mainly consists of a depth-separable convolutional layer (Khan and Niu, 2021), which has fewer parameters and computational effort than traditional convolutional operations, and thus can be used to effectively mitigate the complexity of the model in some resource-constrained scenarios. In addition, depth-separable convolution contributes to enhancing the receptive field, so the DSPE module is employed to substitute for the patch embedding module within the DSFormer, i.e., replacing the conventional convolution with depth separable convolution, which reduces the number of parameters and improves the recognition accuracy.

The depth separable convolution is divided into two parts, depth convolution and point-by-point convolution. For the input characteristic picture of size $H_{in} \times W_{in} \times C_{in}$, the number of convolution kernels is the same as the number of channels for deep convolution, i.e., the size is $K \times K \times C_{in}$, where K denotes the convolution kernel's size, the sliding step of the convolution on the input feature map is S, and the number of pixels that are filled in around the input feature map is P. Therefore, the image with the number of channels C_{in} is computed to produce C_{in} feature maps. The above process is shown in equation (3) and equation (4). After the above process output the dimension of feature map D_{out} as $H_{out} \times W_{out} \times C_{in}$.

$$H_{out} = \left[\frac{H_{in} - K + 2P}{s} \right] + 1 \tag{3}$$

$$W_{out} = \left[\frac{W_{in} - K + 2P}{s} \right] + 1 \tag{4}$$

Point-by-point convolution executes a convolutional operation on every individual pixel point within the input characteristic picture by employing a convolution kernel with a size of D_{out} as $1 \times 1 \times C_{in} \times C_{out}$ and C_{out} is the amount of output channels. During this procedure, the dimensions of the output characteristic picture remain identical to those of the input characteristic picture, the size of the input characteristic picture D_{out} is $H_{out} \times W_{out} \times C_{in}$, and the size of the output feature map is $H_{out} \times W_{out} \times C_{out}$ after point-by-point convolution.

As demonstrated above, depthwise convolution performs convolution function between all channels of the input characteristic picture and the corresponding channel of the convolutional kernel. The resulting characteristic pictures maintain the equivalent amount of channels to that of the input level, thereby limiting characteristic picture

expansion. Consequently, point-by-point convolution is employed to merge these separate feature maps, thereby yielding a novel characteristic picture.

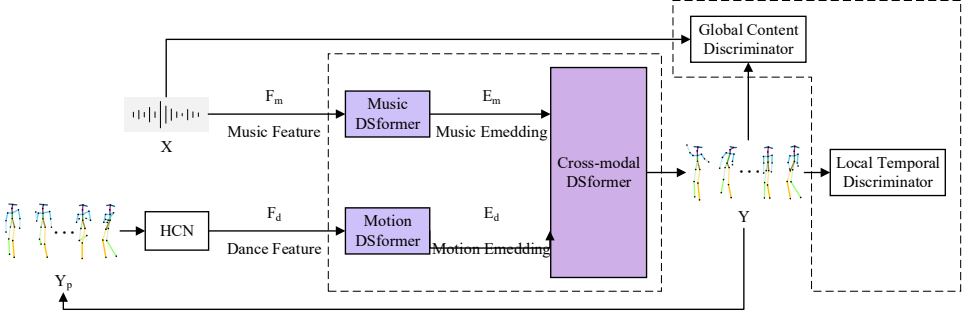
For a 5 × 5 pixel, 3-channel colour input image for depth separable convolution, the first after the depth convolution operation, convolution kernel size of 3 × 3, then the amount of values in the depth convolution section of the calculation is 3 × 3 × 3 = 27. Since point-by-point convolution adopts 1 × 1 convolution, subsequent to point-by-point convolution, four characteristic pictures are yielded, and these characteristic pictures share the same dimensionality as the output generated by regular convolution. The quantity of parameters implicated in the convolution operation during this process is determined to be 1 × 1 × 3 × 4, resulting in a value of 12, so the parameters of the depth – separable convolution are aggregated from two distinct components, yielding a total of 12 + 27 = 39. While the amount of parameters for traditional convolution is 4 × 3 × 3 × 3 = 108, after calculation it can be observed that for the same input, the amount of parameters for depth separable convolution is smaller.

4 Intelligent generation of robotic dance movements based on convolutionally enhanced transformer

4.1 Music encoder

Robotic dance movement generation remains a very challenging task. The poor quality of the generation is due to the noisy data of the joints of the dance movements in the dataset and the fact that existing studies disregard the presence of both local and global dependencies within the sequences of music and dance motions. Therefore, this paper proposes an intelligent generation model for robotic dance movements based on convolutionally enhanced Transformer, the architecture of which is shown in Figure 2, where where music DSFormer, motion DSFormer, cross-modal DSFormer, global content discriminator, motion DSFormer, cross-modal DSFormer, global content and local temporal discriminator are the music encoder, motion encoder, cross-modal generator, global content and native temporal discriminator respectively.

Figure 2 The suggested intelligent generation model for robotic dance movements (see online version for colours)



Given a set of musical features X and an initial robotic dance move or historical dance move Y_P. Firstly, music and motion features are extracted using music and dance motion feature extractor respectively. After that, the music and motion features are encoded as

distributed representation vectors E_m and E_d using music DSFormer and Movement DSFormer, respectively. E_m and E_d are then input into the cross-modal DSFormer for the purpose of producing the following dance actions Y. Final produced robot dance moves will serve as the input data for generating the subsequent dance moves. In addition, this paper also uses local timing discriminators and global content discriminators to improve the coherence of the generated robot dance movements and to make them more consistent with the rhythm and style of the music.

To obtain local spatial features and global temporal features of music sequences, this paper designs a music DSFormer, which is composed of a multi-head attention mechanism, a convolutional layer and a feedforward layer. Unlike DSFormer, to better capture rhythmic temporal correlations, this module applies absolute positional encoding ahead of multi-attention. The music sequence $X = \{x_1, x_2, \ldots, x_t\} \in R^{S\times t}$ is taken as input, and the MFCC feature $F_m = \{f_m^1, f_m^2, ..., f_m^t\} \in R^{M\times t}$ is extracted first. Next add F_m to the absolute position code. Finally, the embedding vectors $E_m = \{e_m^1, e_m^2, ..., e_m^t\} \in R^{d\times t}$, E_m can be obtained by music DSFormer and can be represented by equation (5), in which $ATT(\cdot)$, $DSPE(\cdot)$ and $FFN(\cdot)$ make reference to the multi-head attention scheme, the depth-separable convolutional level and the feedforward layer, respectively.

$$\begin{cases} \tilde{F}_m = F_m + ATT(F_m) \\ \tilde{\tilde{F}}_m = \tilde{F}_m + DSPE(\tilde{F}_m) \\ E_m = \tilde{\tilde{F}}_m + FFN(\tilde{\tilde{F}}_m) \end{cases} \tag{5}$$

4.2 Robot dance motion encoder

Similar to the music encoder, motion DSFormer uses DSFormer to encode robotic dance movement sequences. The module contains a masked multi-head attention mechanism and a convolutional level. It can not only efficiently learn the global features of the dance movement sequences in time series through DSFormer, but also learn the local spatial features through convolutional levels. In terms of position coding, the module combines absolute and relative position coding. Masking operations are performed in the multi-attention mechanism to minimise the impact of noisy robot dance action joints.

For the dance movement sequence $Y_P = \{y_p^1, y_p^2, ..., y_p^t\} \in R^{2V\times t}$, firstly an initial pose needs to be specified for it. Secondly, the motion features $F_d = \{f_d^1, f_d^2, ..., f_d^t\} \in R^{D\times t}$, are extracted by DSPE. Finally the motion features are passed through motion DSFormer to get the embedding vectors $E_d = \{e_d^1, e_d^2, ..., e_d^t\} \in R^{d\times t}$, E_d as shown in equation (6).

$$E_d = F_d + ATT(F_d) + DSPE(F_d + ATT(F_d)) \tag{6}$$

4.3 Cross-modal generator and discriminator

The correlation between music and dance is highly complex and seldom constitutes a deterministic one-to-one correspondence. For the goal of catching the relation between music and dance so that the generated robotic dance movements are chronologically

sequential, this paper designs a cross-modal DSFormer generator, which takes the embedded representations of the music and the historical dance movements as inputs to generate future dance movements. The difference between cross-modal DSFormer and music DSFormer is that cross-modal DSFormer has an additional fully linked level. In contrast to existing cross-modal generators, instead of splicing music and historical dance movement embedded representations, the inputs to the model treat E_d as Q in the MAM and E_m as K and V in the MAM.

E_m and E_d are taken as inputs, and the embedding vector $E_c = \left\{e_c^1, e_c^2, ..., e_c^t\right\} \in R^{d \times t}$ is first obtained by the multi-head attention mechanism, convolutional level and feedforward level. At last, E_c is passed through the fully linked level and the activation operation to gain the future robot dance action sequence $Y = \{y_1, y_2, \ldots, y_t\} \in R^{2V \times t}$ as shown in equation (7), in which $\tanh(LN(\cdot))$ is a single linear level and the tanh operation, and the *FFN* is the feed-forward neural network.

$$\begin{cases} \tilde{E}_d = E_d + ATT\left(E_d, E_m\right) \\ E_c = \tilde{E}_d + DSPE\left(\tilde{E}_d\right) + FFN\left(\tilde{E}_d + DSPE\left(\tilde{E}_d\right)\right) \\ Y = \tanh\left(LN\left(E_c\right)\right) \end{cases} \tag{7}$$

The generated robot dance action sequence Y is classified into K overlapping subsequences, which are then fed into the action discriminator, which is a two-branch convolutional network. The output of the motion discriminator is then used to determine the authenticity of these subsequences using a classifier that outputs K scores.

Whether a robot dance sequence matches the music is a key criterion for evaluating the generated dance sequences, and the model uses a whole content discriminator to capture the global correlation between music and dance. The local content discriminator takes as input the music M along with the robot's dance movement sequence Y. For the robot dance movement section, Y is coded as $F_p \in R^{256}$ using the movement discriminator. For the music part, the music is encoded using a one-dimensional convolutional encoding, and then fed into a bidirectional two-layer GRU to obtain $F_m = AO_m$. Finally, O_m is fed into the self-attention mechanism to obtain the music feature $F_m = AO_m$, where $A = \{a_1, a_2, \ldots, a_n\}$. The expression for a_i is as follows. where W_{s1} and W_{s2} are hyperparameters.

$$\begin{cases} r = W_{s2} \tanh\left(W_{s1} O_m^T\right) \\ a_i = -\log\left(\dfrac{\exp\left(r_i\right)}{\sum_{i=0}^{n} \exp\left(r_i\right)}\right) \end{cases} \tag{8}$$

Finally, F_m and F_p are spliced and passed through a classifier to decide in the event that the sequence of dance movements aligns with the music.

4.4 Joint loss function

The loss of the proposed generative model contains two parts. One part is the DSFormer loss and the other part is the GAN loss. The characteristics captured by DSFormer are high-level spatial structural information between different parts of the robot, and the pre-

trained DSFormer provides better constraints on the details of the movements compared to conventional approaches for example L_1-distance and L_2-distance. In the training phase, we are given a training pair (P, (M, X)). The DSFormer loss can be computed after extracting features using the pre-trained DSFormer as follows.

$$L_p = \sum_{i=0}^{n} \lambda_i \left\| \Phi_i(P) - \Phi_i(G(M, X)) \right\|_1 \tag{9}$$

where P is the real robot dance action sequence, M is the corresponding music clip, X is the initial dance action or past dance action sequence, G is the generator in the GAN, the DSFormer network is denoted by Φ_i, and the hyperparameter λ_i balances the contribution of each level to the loss.

The loss function of GAN is trained based on generators and discriminators. Therefore, the GAN loss is defined as in equation (10).

$$\begin{aligned} L_{GAN} = {} & E_p\left[\log D_{local}(S(p))\right] + E_{x,m}\left[\log\left\{1 - D_{local}(S(x))\right\}\right] \\ & + E_{p,m}\left[\log D_{global}(p, m)\right] + E_{x,m}\left[\log\left\{1 - D_{global}(x, m)\right\}\right] \end{aligned} \tag{10}$$

where D_{local} is the local timing discriminator, D_{global} is the global content discriminator, p is the real dance sequence, m is the corresponding music clip, x is the generated dance sequence, and S is the function that samples the entire dance sequence using a sliding window.

Given a real dance action sequence Y of the same shape as the generated robot dance action sequence X, i.e., the reconstruction loss at the level of the dance action joints is as in equation (11)

$$L_1 = \sum_{i \in [0, 2V]} \| Y_i - X_i \|_1 \tag{11}$$

Feature matching loss: in this paper, feature matching loss is used to stabilise the training of the whole content discriminator D_{global}, as shown in equation (12).

$$L_{fm} = E_{p,m} \sum_{k=1}^{n} \left\| D_{global}^{k}(p, m) - D_{global}^{k}(G(m)), m) \right\|_1 \tag{12}$$

where n is the amount of levels in D_{global} and D_{global}^{k} stands for the k^{th} level in D_{global}.

Finally, the final loss function is obtained by combining equation (9), equation (10), equation (11) and equation (12) as represented in equation (13), where w_p, w_{fm} and w_{l1} denote the weights of each loss term.

$$\arg\min_{G} \max_{D} L_{gan} + w_p L_p + w_{fm} L_{fm} + w_{l1} L_{l1} \tag{13}$$

5 Experimental results and analyses

In this paper, a dataset of music-to-robot dance movements provided in the literature (Alemi et al., 2017) is used to generate the dataset. The dataset contains 45 music and robot videos. Each video is uniformly divided into segments with a base unit of 50 frames

and 1,600 subsequences are obtained. In order to better study the mapping between the music data and the robot's dance movements, the pose movements are represented by a series of key points, ignoring some irrelevant information about the background and the human body. The experiments are realised on RTX-3060 GPU based on PyTorch framework. The DSFormer network proposed in this paper uses a 5-layer stack, and noting that the number of headers is 8. The feedforward layer consists of two fully linked levels, the activation function, and the Dropout level. In this paper, the Adam optimiser is adopted for all networks, where the learning rate of the generator is 0.0004, while the studying rate of both the local timing discriminator and the global content discriminator is 0.0001.

In this paper, firstly, the DSFormer model's are analysed, and the classical Transformer model and its variants Swin Transformer, PoolFormer, and TSN-Transformer are selected for the comparison model, and the outcome of the comparison experiments of different models is indicated in Table 1. The Transformer model has the advantage of modelling global contextual information, but its practical application is limited by the excessively high number of parameters in the multi-head self-attention. Swin transformer has improved the multi-head self-attention to improve the recognition accuracy, but the number of parameters and computational complexity have not decreased to the desired state due to the complexity of the attention module itself. PoolFormer greatly reduces the number of parameters in the model and improves the experimental accuracy of the model by replacing the attention mechanism with a parameterless average pooling layer. TSN-Transformer uses a self-attention mechanism to model features, but its representation of local features is weak. The DSFormer proposed in this paper verifies the effectiveness of DSFormer by introducing parallel max-pooling and average pooling, and replacing regular convolution with depth-separable convolution, which reduces the model parameters by about 60 M and improves the feature extraction accuracy by 4.3% and 2.1%, respectively, compared to Transformer on both datasets.

Table 1 Performance comparison of the transformer model and its variants

Model	*Transformer*	*Swin transformer*	*PoolFormer*	*TSN-transformer*	*DSFormer*
Accuracy/%	81.9	84.6	87.2	88.1	90.7
Number of parameters	75.3	62.7	12.4	11.6	9.7

To further validate the effectiveness of the DSFormer model for robot dance movement generation, this paper selects Frechet inception distance score (FID), beat align score (BAS), beat coverage (BC), beat hit rate (BHR), structural similarity (SSIM), learnable perceptual image block similarity (LPIPS) as evaluation metrics. Comparison experiments are conducted on DSFormer as well as the benchmark models COV-LSTM (Liu and Ko, 2022), STGAN (Liang, et al., 2024), and TRANS-LSTM (Sun and Wang, 2024), and the FID comparisons of the different models are shown in Figure 3. FID is an important metric for evaluating the quality of generation; the lower the FID, the closer the two distributions are, which means that the generated action sequences are more similar to the real action sequences. The FID curves of STGAN and TRANS-LSTM are steeper and fluctuate significantly. This is due to the accumulation of errors in long sequences, resulting in lower quality dances. The COV-LSTM and DSFormer models

have slow-growing and smoother FID curves, but the FID value of DSFormer stabilises at 31.5 after many rounds of iterations, which is at least 21.64% lower compared to COV-LSTM, STGAN, and TRANS-LSTM, indicating that DSFormer has the highest quality of dance movement generation.

Figure 3 FID comparisons of the different models (see online version for colours)

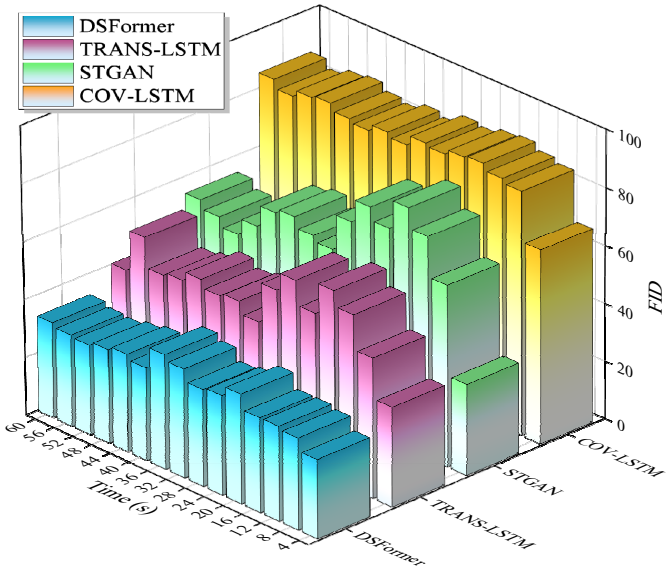
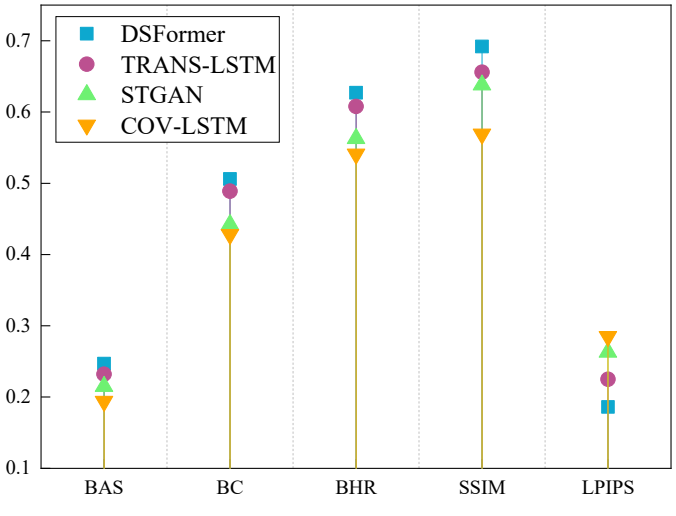


Figure 4 Comparison of robotic dance movement performance metrics (see online version for colours)



Comparison of BAS, BC, BHR, SSIM and LPIPS for different models is shown in Figure 4. The BAS, BC, and BHR of DSFormer are 0.247, 0.506, and 0.627, respectively, which are improved by 27.32%, 18.22%, and 15.89% compared to COV-LSTM, 14.88%,

14.48%, and 11.37% compared to STGAN, and TRANS-LSTM by 6.47%, 3.48%, 3.13%. DSFormer has a SSIM of 0.692, which is a 5.49%-21.62% improvement over the other three models. LPIPS is used to measure the difference between two images.The LPIPS of DSFormer is minimised, which indicates that the difference between the robot dance movement images generated by DSFormer is small and the quality of the generated images is high. DSFormer employs DSFormer encoder and Cross-modal DSFormer generator to perform better than COV-LSTM, STGAN, TRANS-LSTM models. The DSFormer encoder is better able to capture local spatial features and global features in the time series, while the cross-modal DSFormer generator is better able to establish correlations between music and dance. As a result, DSFormer was able to harmonise the generated robotic dance sequences with the music and outperformed the benchmark model in all evaluation metrics.

6 Conclusions

This paper proposes a method for intelligent generation of robot dance movements based on convolutionally enhanced Transformer. Firstly, to address the problem of large number of parameters in the Transformer multi-head self-attention module, the Transformer network is enhanced based on depth-separable convolution. DSFormer takes advantage of the linear complexity of pooling with respect to the length of the sequence as well as the parameterlessness, and introduces a parallel pooling module instead of a multi-headed self-attention module, which effectively decreases the number of parameters while improving the computational efficiency of the model. Second, the depth-separable convolutional module is used to replace the traditional patch embedding module, which further reduces the number of parameters while improving the calculational efficiency without losing the performance of the model. On this basis, the music encoder, robot dance movement encoder and cross-modal generator are constructed based on DSFormer, which captures the local spatial features and global features of the music and robot dance movement sequences in the time series, so as to be able to catch local and global dependencies between the sequences and to decrease the influence of noisy data. Simulation outcome indicates that the SSIM of the proposed method is 0.692, which is improved by 5.49%–21.62% compared to the other three methods, and can generate higher quality dance movements.

The intelligent generation model of robotic dance movements proposed in the paper leaves much to be desired in terms of diversity and beat alignment. In future work, this paper will introduce a reinforcement learning module to enhance the model's understanding of music and dance beats, and to generate more coherent and natural sequences of robotic dance movements. Finally, a real-time rendering system will be implemented to further extend the application of the model.

Declarations

All authors declare that they have no conflicts of interest.

References

Ahn, H., Kim, J., Kim, K. and Oh, S. (2020) 'Generative autoregressive networks for 3d dancing move synthesis from music', *IEEE Robotics and Automation Letters*, Vol. 5, No. 2, pp.3501–3508.

Alemi, O., Françoise, J. and Pasquier, P. (2017) 'GrooveNet: Real-time music-driven dance movement generation using artificial neural networks', *Networks*, Vol. 8, No. 17, pp.13–26.

Cong, S. and Zhou, Y. (2023) 'A review of convolutional neural network architectures and their optimizations', *Artificial Intelligence Review*, Vol. 56, No. 3, pp.1905–1969.

Friedman, D., Wettig, A. and Chen, D. (2023) 'Learning transformer programs', *Advances in Neural Information Processing Systems*, Vol. 36, pp.49044–49067.

Han, B., Li, Y., Shen, Y., Ren, Y. and Han, F. (2024) 'Dance2MIDI: Dance-driven multi-instrument music generation', *Computational Visual Media*, Vol. 10, No. 4, pp.791–802.

Khan, Z.Y. and Niu, Z. (2021) 'CNN with depthwise separable convolutions and combined kernels for rating prediction', *Expert Systems with Applications*, Vol. 170, pp.45–58.

Kritsis, K., Gkiokas, A., Pikrakis, A. and Katsouros, V. (2022) 'Danceconv: dance motion generation with convolutional networks', *IEEE Access*, Vol. 10, pp.44982–45000.

Lee, J.-S. and Lee, S.-H. (2019) 'Automatic path generation for group dance performance using a genetic algorithm', *Multimedia Tools and Applications*, Vol. 78, No. 6, pp.7517–7541.

Li, J., Peng, H., Hu, H., Luo, Z. and Tang, C. (2020) 'Multimodal information fusion for automatic aesthetics evaluation of robotic dance poses', *International Journal of Social Robotics*, Vol. 12, pp.5–20.

Liang, X., Li, W., Huang, L. and Gao, C. (2024) 'DanceComposer: dance-to-music generation using a progressive conditional music generator', *IEEE Transactions on Multimedia*, Vol. 26, pp.10237–10250.

Liu, M., Shi, J., Li, Z., Li, C., Zhu, J. and Liu, S. (2016) 'Towards better analysis of deep convolutional neural networks', *IEEE Transactions on Visualization and Computer Graphics*, Vol. 23, No. 1, pp.91–100.

Liu, X. and Hu, J. (2021) 'Dance movement recognition technology based on multifeature information fusion', *Journal of Sensors*, Vol. 20, No. 1, pp.79–87.

Liu, X. and Ko, Y.C. (2022) 'The use of deep learning technology in dance movement generation', Frontiers in Neurorobotics, Vol. 16, pp. 23-37.

Ma, X., Zhang, P., Zhang, S., Duan, N., Hou, Y., Zhou, M. and Song, D. (2019) 'A tensorized transformer for language modeling', *Advances in Neural Information Processing Systems*, Vol. 32, pp.21–37.

Peng, H., Hu, J., Wang, H., Ren, H., Sun, C., Hu, H. and Li, J. (2021) 'Multiple visual feature integration based automatic aesthetics evaluation of robotic dance motions', *Information*, Vol. 12, No. 3, pp.83–95.

Peng, H., Li, J., Hu, H., Hu, K., Zhao, L. and Tang, C. (2022) 'Automatic aesthetics assessment of robotic dance motions', *Robotics and Autonomous Systems*, Vol. 155, pp.41–54.

Qin, R., Zhou, C., Zhu, H., Shi, M., Chao, F. and Li, N. (2018) 'A music-driven dance system of humanoid robots', International *Journal of Humanoid Robotics*, Vol. 15, No. 5, pp.18–27.

Sun, D. and Wang, G. (2024) 'Deep learning driven multi-scale spatiotemporal fusion dance spectrum generation network: a method based on human pose fusion', *Alexandria Engineering Journal*, Vol. 107, pp.634–642.

Valle-Pérez, G., Henter, G.E., Beskow, J., Holzapfel, A., Oudeyer, P.-Y. and Alexanderson, S. (2021) 'Transflower: probabilistic autoregressive dance generation with multimodal attention', *ACM Transactions on Graphics* (*TOG*), Vol. 40, No. 6, pp.1–14.

Vonder Haar, L., Elvira, T. and Ochoa, O. (2023) 'An analysis of explainability methods for convolutional neural networks', *Engineering Applications of Artificial Intelligence*, Vol. 117, pp.38–47.

Wang, H., Song, Y., Jiang, W. and Wang, T. (2024) 'A music-driven dance generation method based on a spatial-temporal refinement model to optimize abnormal frames', *Sensors*, Vol. 24, No. 2, p.588.

Xu, F., Xia, Y. and Wu, X. (2023) 'An adaptive control framework based multi-modal information-driven dance composition model for musical robots', *Frontiers in Neurorobotics*, Vol. 17, pp.12–34.

Zhou, D.-X. (2020) 'Theory of deep convolutional neural networks: Downsampling', *Neural Networks*, Vol. 124, pp.319–327.

Zhou, Z., Huo, Y., Huang, G., Zeng, A., Chen, X., Huang, L. and Li, Z. (2024) 'Qean: quaternion-enhanced attention network for visual dance generation', *The Visual Computer*, Vol. 41, pp.1–13.

Zhuang, W., Wang, C., Chai, J., Wang, Y., Shao, M. and Xia, S. (2022) 'Music2dance: Dancenet for music-driven dance generation', *ACM Transactions on Multimedia Computing, Communications, and Applications* (*TOMM*), Vol. 18, No. 2, pp.1–21.