International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 17 (2025) pp. 1-21

https://doi.org/10.70917/ijcisim-2025-0304
Article

A Study on Generative Model-based Dance Movement Creation and Virtual Dancer Generation

Yiheng Li *

Department of Dance, Fujian Vocational College of Art, Fuzhou, Fujian, 350100, China； L_yyhh@163.com

Abstract: With the development of digital entertainment, obtaining intelligent dance creation under musical conditions is currently a popular research field. Traditional dance creation has the problems of long creation cycle and limited creative inspiration. This study proposes an end-to-end generative framework to realize automatic generation from music to dance movements with real-time driving of virtual dancers. Firstly, the dance data is obtained by motion capture technology, and polygonal modeling technology is used for virtual character model construction. Then the audio and motion features are obtained from the audio signal and human body motion sequences respectively. In which the core section adopts TransGAN-based intelligent dance generation network, which uses generative adversarial network as a framework to combine Transformer and up-sampling layers to realize multi-level action coding. Experiments show that the coherence and rationality of the dance movements generated by this model are higher than the comparison model. The error of the hip joint of the virtual dancer's walking movement is only 1.006, which is able to realize stable, accurate and diverse dance performance. This study provides a feasible technical solution to promote the automated production of dance content.

Keywords: Dance movement generation; TransGAN; Motion capture technology; Virtual human driving algorithm

1. Introduction

Choreography is a comprehensive form of artistic expression, the essence of which lies in conveying emotions, ideas and stories through elements such as body movements, gestures and rhythms [1-2]. Its core elements include the dancer's body language, the cooperation of music, the use of space, and the injection of emotion. With the development of artificial intelligence, the application of generative modeling in dance creation has changed the process of dance creation and brought new technological tools for dance performance [3-5].

Generative model is a kind of machine learning model used to generate similar to the observed data, by learning the potential representation and distribution of real data, it can generate new and similar to real data, which is widely used in the field of image, speech, text and so on [6-9]. Its important application in dance movement creation is dance and movement recognition. Generative models can accurately analyze dancers' movements and poses and convert them into digital data, which can be used to analyze the dancers' skill level, the accuracy of their poses, and the fluidity of their dance movements, thus helping dancers and choreographers to improve their acting and creation [10-13]. In addition, generative modeling can generate new dance movements and choreography in dance creation [14]. Generative models can generate new movement sequences and choreography by learning and understanding the fundamentals and styles of dance [15-16]. This generative process can be guided by inputting different parameters, such as emotion, rhythm, and style, to guide the human generative model in generating different styles of dance movements, which provides choreographers with a new creative tool and idea, making the process of dance creation richer and more diverse [17-20]. In addition to dance movement creation, the generative model is also capable of generating virtual dancers. By capturing and analyzing the dance movements, the generative model can generate realistic virtual characters and realize

the interaction with real dancers [21-22]. The virtual characters can learn and evolve independently through artificial intelligence technology, continuously improve their dance skills, and carry out perfect collaboration and dialog with the dancers [23].

In the era of artificial intelligence, in the field of dance movement generation, in addition to generative models, deep learning, neural networks and other technologies and models also play an important role and have their own characteristics. Literature [24] pointed out the shortcomings of traditional methods for generating dance movements, and proposed a deep learning-based dance movement generation method, which effectively generates realistic dance movements by extracting the mapping relationship between sound and movement features. Literature [25] proposed the application of multimodal convolutional self-coder in dance movement generation, which is able to generate dance movements of arbitrary length by combining 2D skeletal information and audio information, and exhibits realism, diversity and other characteristics. Literature [26] proposes a bidirectional autoregressive diffusion model for music-to-dance generation and verifies that the model achieves state-of-the-art performance in a prestigious benchmark test for music-to-dance generation compared to existing unidirectional methods. Literature [27] describes a deep learning-based consistent movement generation model for dance that is capable of generating long sequences of dance movements with coherent patterns through latent space interpolation, demonstrating excellent generation accuracy, precision and recall. Literature [28] examined the requirements and system design of a recurrent neural network-based dance generation system, and completed a dance generation algorithm by combining the latest image generation techniques as well as open-source gesture detection projects, which played an important role in generating dance movements. Literature [29] proposes a method for automatic generation of folk dance movements, which is used in practice to verify that it performs well in the automatic generation of folk dances, and the generated dances are characterized by folk characteristics and can be matched with music rhythms. Literature [30] developed a deep generation model that combines movement generation and style conversion, which not only can achieve accurate conversion between different styles of dance, but also performs well in generating dance movements with smooth and colorful movements. Literature [31] developed a real-time virtual reality dance training framework using an augmented converter model, which can generate partner movement sequences based on user movements, and has important application prospects for intelligent dance teaching and human-computer interaction.

Current mainstream dance movement creation methods mainly rely on simple recurrent neural networks, and these methods have serious deficiencies in terms of movement innovativeness and coherence of complex movements. To overcome these deficiencies, this study explores the application of generative adversarial networks based on the Transformer framework to the task of dance movement generation. The model utilizes Transformer's advantage in capturing long sequence dependencies, combined with the adversarial training mechanism of GAN, to ensure a high degree of matching between dance movements and music. Finally, based on the virtual human-driven algorithm to stitch the generated movement data with the virtual character, the movement performance of the virtual dancer is verified in a complex simulation environment, realizing the whole process automation from audio input to virtual dancer performance.

2. Generative model-based dance movement creation and virtual dancer generation

2.1. Dance Motion Capture

The Vicon optical motion capture system is the world's first optical motion capture system designed for animation production with excellent motion capture performance. Therefore, Vicon was selected to capture dance movements in this study, and its basic process is shown in Figure 1. First, the actors are selected and choreographed, and the marker locations are determined according to ergonomic principles, and the marker models are built using ViconShōgun. Then use Maya software to build character models and create character model costumes. Meanwhile, in order to realize efficient motion capture and generate high-quality character models, neural fusion shape technology is used to generate bones and bind bone skin weights in order to automatically generate high-quality character models. Finally, the data fusion algorithm is used to complete the driving of finger movement in the character model and synthesize the character model for export, i.e., motion capture is realized. Based on the above ideas, the research mainly designs motion capture from three parts: marker point labeling and processing, character model construction, and motion driving.

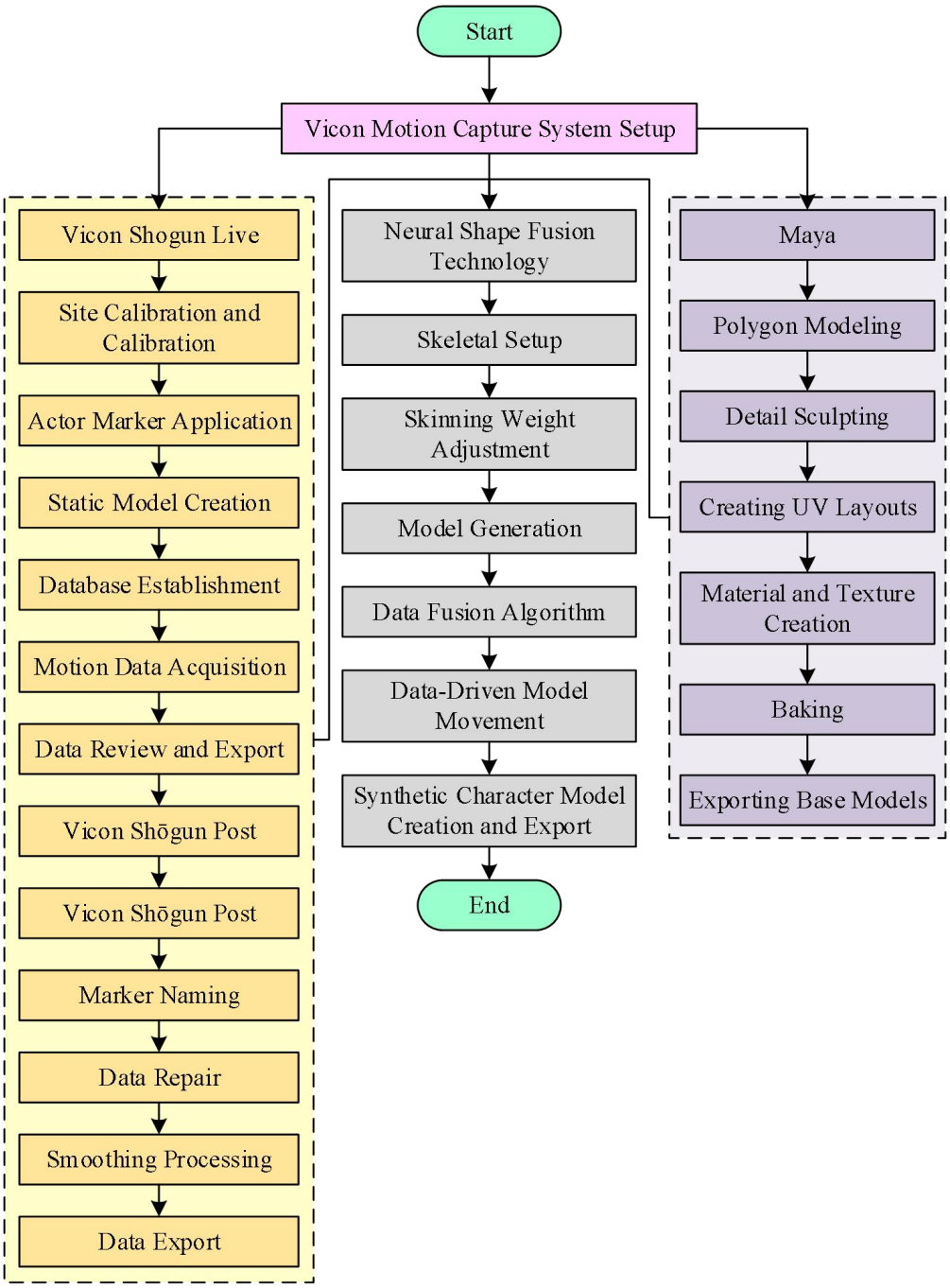

Figure 1. Action capture process.

2.1.1. Marker point labeling and processing

Combining ergonomic principles and dance characteristics, 60 marker points are selected to be pasted on the whole body of the actor. Considering that some marker point locations may be obscured in the dance movements, Vicon Shōgun is used to solve the obscured marker points and build a model to capture the dance movements. At the same time, the problematic marker points and locations are viewed and repaired based on Shōgun post. Finally, the skeleton is solved and the output data is the complete human marker points.

2.1.2. Constructing virtual character models

Maya animation production software is used for virtual character model construction. Specifically

using polygonal modeling technology to model the head and body of the dancer; then composed of a complete human body model; and finally adjusted and modified the body parts of the stitching and details, and finally realized the complete character model construction.

On the basis of character modeling, the creation of clothing is carried out. First of all, we create clothes sample curves before and after the body of the character model; then we simulate the material and texture of the real fabric through the 2D texture method; finally, we adjust the editing UV and texture editor to apply the texture to the corresponding objects, thus realizing the creation of the character's clothing model.

The character model consists of a large number of vertices, which will increase the workload if each frame is moved to the specified position manually. Therefore, key skeletal nodes are extracted to control the nodes, and then the character model is constructed. And to realize data-driven character models, bone building and skin weight binding are also required. To further simplify the process, referring to the experience of some researchers, the neural fusion shape technique is used to generate bones and bind the skeleton skin weights. The basic flow of the neural fusion shape technique is shown in Figure 2.

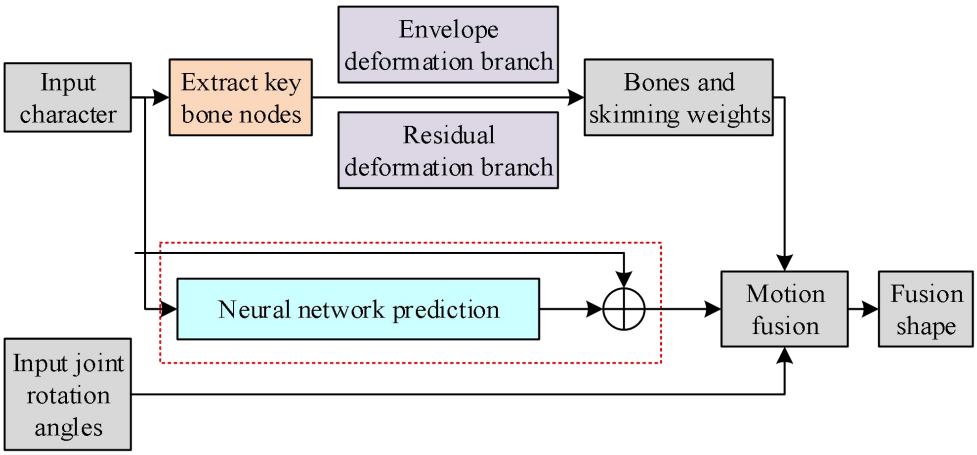

Figure 2. Neural fusion shape technology.

The neural fusion shape technology framework mainly includes envelope deformation branch and residual deformation branch. By inputting the joint rotation and character model data, the skeletal structure can be obtained; then the envelope deformation branch is used to bind the skin weights of the skeleton; finally, it is fused with the hybrid shapes and hybrid coefficients in the residual deformation branch, and the fused character model is thus obtained.

(1) Envelope deformation branch

This part focuses on learning the parameters of a specific skeleton level and predicting the skeleton, skin and weight bindings. A pose $R = \{R_i\}\ where\ R_i \in R^{3\times3}$ represented by a local joint rotation is added in each iteration, and the deformation of the input data and the predicted bindings and skinning are guided by this value. Therefore, it is necessary to first perform a local mapping transformation $\{R_i, O_i\}$ for each joint in the character model using the positive kinematic cumulative transformation; and then compute a global transformation for each vertex according to Eq. (1):

$$T_{R_j} = \sum_i W_{ji} T_i \tag{1}$$

Eq. (1) is the mask matrix calculation formula. The vertex-by-vertex mapping transformation $T_R = \{T_{R_i}\}$ can be obtained by applying it to the input roles after the operation:

$$\tilde{V}_{R_j} = T_R \Theta V \tag{2}$$

In Eq. (2), T_R is the global mapping transformation and Θ is the vertex-by-vertex operation.

(2) Residual deformation branching

Let the given vertex position be V and the connectivity of each vertex be F, the output skin W is connected to the depth vertex V' along the channel $\left(\{V', W\} \in R^{1\times(K+J)}\right)$. Then a set of N residual

shapes $\{B_i\}_{i=1}^{N}, B_i \in R^{1\times 3}$ is generated using the edge feature representations and mesh convolution width. A small neural network of J MLP blocks is input simultaneously, and finally the pose-dependent coefficients $\{a_{ij}\}_{i=1}^{N}$ of each joint J are output and added to the residual shape of V. This is specifically represented as:

$$\tilde{V} = V + \sum_{j=1}^{J} \sum_{i=1}^{N} \alpha_{ij} M_j B_i \quad (3)$$

In Eq. (3), M_j is a binary mask that specifies the vertices associated with joint j.

Finally, the loss function L_e is utilized to find the difference between the task roles and the corresponding true values.

As a result, high-quality 3D character models are automatically generated by the neural fusion shape technique of envelope deformation branches and residual deformation branches.

The above fusion is only used for the torso, limbs, etc., but for modern dance, the most critical is the hand movement. Therefore, it is also necessary to drive the bones of the finger details. In this regard, it is proposed to use data fusion algorithm to complete the driving of finger movement in the character model. The specific process is as follows.

(1) Read the data in the motion capture file.

(2) Read the information in the model file.

(3) Match the skeletal nodes of the two files, and according to the matching result, read the position of the corresponding nodes in the motion capture file, and make the nodes in the model display in the node position of the motion capture file. Finally, the nodes in the model are matched with the nodes in the motion capture file.

(4) Repeat the cycle of step (4) to completely integrate the task model to the captured action sequence, so as to realize the effect of motion capture data driving the movement of the character model.

2.2. Audio Representation and Feature Extraction

Currently most of the audio is saved in file formats such as MP3, MP4, WAV, etc., which must be digitized in order to be recognized and utilized by the model, and the extraction of audio features determines the learning ability of the model as well as the generation of actions. The following is a specific description of the representation of audio and the process of feature extraction.

2.2.1. Audio processing

Sound arises from changes in air pressure, while audio signals are used to represent changes in sound as a waveform signal. According to the waveform can be divided into regular and irregular signals, regular signals belong to analog signals, which are characterized by continuous and regular. Common regular signals are music, speech and so on. In life, most of the music we hear belongs to analog signals, however, the model network can not use analog signals as inputs, so it must be converted into digital signals that can be processed, and the conversion process includes sampling, quantization and encoding.

2.2.2. Audio Characterization

Although the processed audio files can be recognized by the computer, for deep modeling, some regular features are needed as input for better model training. According to the different ways of extraction, audio features can be divided into the following categories:(1) Global features and local features. (2) Statistical features and output features. (3) Bottom-level features and high-level features. Common high-level audio features can include time domain, energy, musicality, frequency domain, and perceptual features.

2.2.3. Audio Feature Extraction Methods

The common methods used for feature extraction for audio are convolutional neural network extraction and feature tool extraction. Using different methods, the extraction results are different. For the former, usually only the features between digital signals can be obtained, but not the deep features of the audio, and the extraction process is relatively complex. For the latter, the extracted features are richer and the operation is simple.

(1) Convolutional Neural Network

Convolutional neural network [32] is a commonly used feature extraction network, consisting of

convolutional layer, pooling layer, fully connected layer, usually used for image processing. For audio digital signals, it is converted to a regular matrix, which can be used to extract features with CNN.

Audio: In image processing, the input picture usually has R, G, and B3 channels, and the picture is represented by a $W*H*C$ three-dimensional matrix, which represents the width, height, and number of channels, respectively. Similar to pictures, the same RGB matrix can be used to represent the digital signal of audio. Before inputting the convolutional layer, a fill operation can be done on the audio to change its dimensions.

Convolutional Layer: It is used for feature extraction of audio.

Let the width and height of the digital signal be W and H respectively, the size of the convolution kernel be D1, D2, the step size of each move be Stride, and the image padding be Padding, the dimension of the extracted features can be calculated as:

$$out_h = \frac{W - D1 + 2*padding}{stride} + 1 \tag{4}$$

$$out_w = \frac{W - D2 + 2*padding}{stride} + 1 \tag{5}$$

Pooling Layer: the role of the pooling layer is to filter the features. This screening improves the robustness of the model and prevents overfitting. Constant

Fully Connected Layer: the role of the Fully Connected Layer is to change the dimensionality of the output features so that they satisfy the specific modeling task.

Extracting audio features through convolutional neural networks is relatively complex because the structure of the feature extraction network needs to be carefully designed, and due to errors in the network computation, it needs to be trained several times to extract a better representation of the features.

(2) Tool Extraction

Feature extraction tools based on Python/C language can effectively and quickly help researchers to extract audio features, and different tools differ in feature extraction. In this paper, we utilize LibROSA to process music files. Users can install the LibROSA library directly using the command, and when using it, they only need to input a piece of audio file (WAV or MP3 format) and its sample rate, and then they can call the corresponding algorithm to calculate the features according to their needs.

2.3. Action Representation and Feature Extraction

The purpose of action feature extraction is to capture the correlation between action frames and the connection of nodes within the action. In addition, the action feature distance is an important evaluation metric. This section describes different types of dance action representations and feature extraction networks.

2.3.1. Motion Representation

In the field of music generation and dance, movements are usually presented in the form of human skeleton, and the common movements in the current publicly available datasets can be categorized into 2D movement datasets and 3D movement datasets according to their representation. In 2D datasets, movements are generally represented by 2D coordinates or polar coordinates of each joint point, which has the advantage of being simple to extract and easy to learn, and the disadvantage of lacking expressive power.3D movement datasets have a relatively complex extraction process, but the human body movements are more vivid, and there are various forms of representation.

2.3.2. Motion Feature Extraction Methods

Action is a graph structure, although the generated action is saved in the form of a picture, its essence is the coordinates of the joint points between the skeletons are drawn using a drawing tool, so it is difficult for traditional image feature extraction networks to accurately extract the action features. Common extraction methods include graph convolutional neural network, spatio-temporal graph convolutional neural network and so on.

(1) Graph Convolutional Neural Network

A graph consists of nodes and edges, where nodes contain intrinsic information and edges contain relationships between nodes. Graph Convolutional Neural Network (GCN) [33] is designed based on graph structure and can directly extract graph features. It is usually used in tasks such as node classification and link prediction.

During feature extraction, the graph convolutional network traverses each node in the graph in turn.

For a given moment, the network feature output is calculated as:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \tag{6}$$

Since the number of neighboring nodes around each graph node in the input graph data is variable, the number of nodes computed during sampling varies. The self-connecting neighbor matrix and degree matrix are similar to the convolution kernel of a convolutional neural network that samples the nodes according to some rule.

The role of the self-connecting adjacency matrix is to update the input features, and the role of the degree matrix is to weight the updated features so that for a given node, if its degree is higher, it contains less information.

The updated features continue to be used as inputs to the next layer and the node features are updated again. After multiple layers of graph convolution, the graph convolution network can flexibly capture the relationship between the nodes.

(2) Spatio-temporal graph convolutional neural network

Spatiotemporal graph convolutional network (ST-GCN) [34] contains graph convolutional network (GCN) and temporal convolutional network (TCN), which represents the human body action as a graph with nodes corresponding to the human body joints, and edges are categorized into spatial edges connecting the two neighboring nodes within a single frame of the action, and temporal edges connecting the same nodes in multiple frames of the action. Spatial features within a single frame are extracted using graph convolution, and temporal features between multiple frames are extracted using temporal convolution, which achieves excellent results in action feature extraction.

In the process of graph convolution, in order to learn the correlation between different joints during S extraction, ST-GCN proposes three partitioning strategies.In this paper, we use spatial configuration partitioning: the center of gravity of the current action is computed, and it is divided into central nodes, centripetal clusters, and centrifugal clusters according to the distance from the center of gravity.

2.4. TransGAN-based dance movement generation

Compared to the 2D image generation task which is also a visual task, the dance generation task in 3D has more challenges, for this reason this section proposes a TG-dance model for generating 3D dance movements under musical conditions.The TG-dance model is shown in Fig. 3.

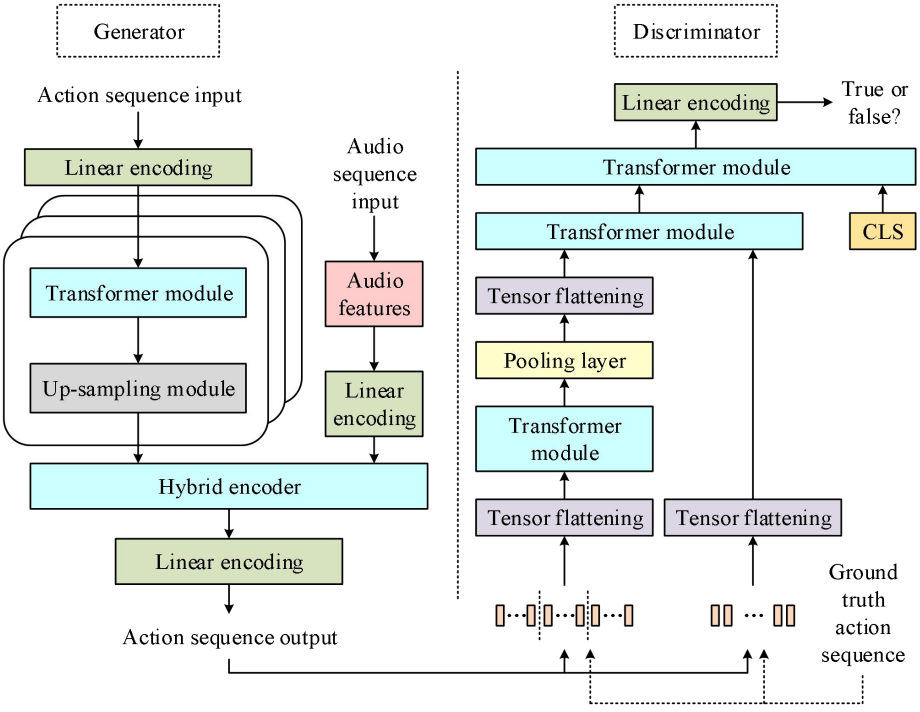

Figure 3. TG-dance network structure diagram.

In order to investigate Transformer even further on visual tasks, the feasibility of designing a pure architecture consisting of Transformer only was explored. An attempt was made to design a new generative model, TransGAN, with GAN as the backbone architecture and all modules using only Transformer structures. TransGAN has superior performance compared to the state-of-the-art GAN using convolution. Transformer and GAN try to combine models that can solve the problem of image generation tasks by upsampling multilevel scaling to high resolution generation.

2.4.1. Generator module

Due to the complexity of human movement, music and action were not fused early on, but rather the action sequences were first learned separately in multiple phases, and music conditions were subsequently introduced to learn the correlation between the two. The generator module contains three parts: action encoder, music feature extractor and hybrid encoder. The action encoder uses a combination of the Transformer module and the upsampling module, the music feature extractor is used to extract multiple features from the music, and the hybrid encoder uses both the action and the music as inputs and learns them together.

(1) Transformer module

The recent new structure Transformer has been widely used in multi-domain tasks. Compared with convolutional neural networks, Transformer completely abandons recursion and convolution, and uses only the attention mechanism, which makes it easier to obtain contextual information. A large number of studies have proved the feasibility of the Transformer structure and it has been widely used in the field of natural language processing, and has become a hot structure in the research of generative tasks because of its good performance in sequential tasks. In order to take advantage of Transformer's own advantages, the generator structure takes Transformer as the basic module, and the structure includes four parts: positional coding layer, normalization layer, multi-head self-attention layer, and feed-forward layer.

In order to ensure that the relationship between the input sequences is not lost, the position of the input sequences is first encoded using the position encoding layer, which labels each sequence with unique information by adding absolute position values. The position encoding (PE) is calculated using the following function equation:

$$PE(pos,2i)=\sin\left(\frac{pos}{10000^{\frac{2i}{d_{\mathrm{mod}el}}}}\right) \tag{7}$$

$$PE(pos,2i+1)=\cos\left(\frac{pos}{10000^{\frac{2i}{d_{\mathrm{mod}el}}}}\right) \tag{8}$$

The Multihead Attention [35] layer runs the multiple attention mechanism in parallel and captures richer features compared to self-attention. This layer employs 4 heads and is computed as follows:

$$\begin{aligned} &MultiHead\left(Q,K,V\right)=Concat\left(head_1,...,head_h\right)W^O\\ &Other\ head_i=Attention\left(QW_i^Q,KW_i^K,VW_i^V\right) \end{aligned} \tag{9}$$

After the inputs are transformed into subqueries, subkeys, and subvalues, each header is independently subjected to a linear transformation and scaled dot product attention operation, computed as follows, where d_k is the dimension of the input:

$$Attention\left(Q,K,V\right)=soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)\cdot V \tag{10}$$

The feed-forward layer uses Gaussian Error Linear Units (GELUs) as the activation function of the multilayer perceptron (MLP), and the GELU formula is shown below:

$$GELU(x)=xP(X\le x)=x\Phi(x)=x*\frac{1}{2}\left[1+erf\left(\frac{x}{\sqrt{2}}\right)\right] \tag{11}$$

Layer normalization (LN) operations were performed before the attention and feedforward layers,

respectively. The formula is calculated as follows, where $E(x)$ is the mean of the input x, $Var(x)$ is the standard deviation of the input x, and γ and β are the parameters of the learnable affine transformation:

$$y = \frac{x - E(x)}{\sqrt{Var(x) + \text{ò}}} * \gamma + \beta \tag{12}$$

The summation process after the attention and feedforward layers belongs to the residual network. Residual network is to add the input term x with $F(x)$ obtained by passing through the function F, this way can solve the problem of deep neural network degradation and has a faster convergence speed.

(2) Upsampling modules

The field of dance generation can be categorized into two approaches based on whether or not action sequences are introduced as one of the inputs. Some experiments have demonstrated that inputting some of the action sequences into the network can learn more reasonable action generation. This module realizes the reduction of the number of frames of the input action sequences through multiple up-sampling operations. The action sequence input is first passed through a linear transformation (Linear) layer outputting a size of $l_m \times C$, where l_m is the number of frames in the input sequence and C is the hidden layer value. The Transformer module learns the features of the input action sequence and outputs a size of $l_m \times C$.

The action sequences are passed through an action encoder to obtain a long sequence of action features, and in order to fuse them with the music, a hybrid encoder is used to learn the correlation between the two. The hybrid encoder firstly encodes the action feature input through a layer of Transformer module for further encoding, and the music feature input is encoded with position before, these two modalities are spliced together and then encoded uniformly through the hybrid Transformer structure to obtain the output, where the hybrid Transformer structure used is the same as that used in the action encoder.

The loss function of the generator uses a combination of L1 loss, MSE loss and BCE loss.The formulae for L1 loss and MSE loss are given below:

$$loss_{L_1}(x, y) = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - f(x_i)\right| \tag{13}$$

$$loss_{MSE}(x, y) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 \tag{14}$$

BCE loss is one of the frequently used loss functions for classification tasks, where the generated action sequences and the real action sequences go through a discriminator to get the classification results, and the BCE loss is used to calculate the gap between the two:

$$loss_{BCE}(p, t) = -w\left[t\log(p) + (1-t)\log(1-p)\right] \tag{15}$$

where p is the model prediction, t is the true label value, and w is the weight. When the classification result is closer to 1 indicates that the discriminator thinks that the possibility of real samples is greater, while the classification result is closer to 0 indicates that the discriminator thinks that the possibility of generated samples is greater. In order to avoid the discriminator's overclassification ability, the parameters of the above three losses are set to 1:1:0.01 during the training process.

2.4.2. Discriminator module

The generator predicts future dance action sequences under the input conditions of action sequences and audio features. In the generative adversarial network, the discriminator is used to make a true or false judgment between the generated action sequences and the real action sequences, and the capability of the generator is improved by continuous confrontation with the discriminator, which in turn improves the quality of the generated actions. Before inputting the discriminator, the action sequence is convolved in two dimensions to generate a two-part action feature sequence, which is calculated as follows:

$$L_i = \left(\frac{\sqrt{L_{in}} + p_i - d_i * (k_i - 1) - 1}{s_i} + 1\right)^2 \tag{16}$$

Where L_{in} is the length of the input sequence, L_i is the length of the i th output sequence, p_i is the padding size at the boundary; d_i is the distance between the control points; k_i is the size of the convolution kernel; and s_i is the step size of the convolution operation.

The discriminator structure is designed with a three-stage Transformer module. Firstly, tensor flattening operation is performed on the above two parts of feature vectors respectively to obtain the sequences S_A and S_B, and then the sequence S_A is input to the first Transformer module for encoding, followed by one pooling and tensor flattening operation to obtain the sequence $S_{A'}$, which is spliced with the sequence S_B and input to the second Transformer module for encoding to obtain the sequence S_C.

2.5. Virtual Human Driving Algorithm

After the virtual human model has been established, this section drives the virtual human model according to the existing motion data. The human body structure is organized in a tree hierarchy, the root node of the tree is Root and other nodes correspond to the joints of the virtual human, each joint has a certain degree of rotational freedom (1 to 3), each joint rotates around its own axis of rotation, and at the same time, the motion of the parent joint will drive the motion of the child joints, so this paper adopts a hierarchical approach to describe the motion.

The motion states of the virtual human can be categorized into translation and rotation, i.e., translation of the Root point and rotation of each joint point due to its parent node drive or rotation around the respective rotation axis. Translation is expressed in 3D coordinates, i.e., the coordinate position of each joint point in 3D space is recorded. Rotation can be expressed in terms of a rotation matrix, Euler angles, or quaternions, where a rotation matrix is a matrix that acts on a vector in such a way that it only changes the direction of the vector but not its magnitude, and an Euler angle is a set of three independent angular coefficients used to uniquely determine the position of a rigid body rotating at a fixed point and consists of the chapter angle θ, the angle of progression ψ, and the angle of rotation ϕ. A quaternion is a four-dimensional vector: $Q = \langle w, x, y, z \rangle$, with the following complex form: $Q = w + xi + yj + zk$. In graphics, the unit quaternion is often used to describe the rotation of an object in three-dimensional space, in the geometric sense that the object is rotated about the axis $\langle x, y, z \rangle$ by an angle which is expressed as $\cos(\theta / 2) = w$.

In this paper, $P_i^t(x, y, z)$ is used to denote the position of node i under the global coordinate system at the moment t, where when i is Root it denotes the position of the root node, when $1 \le i \le \text{num}$ (num is the number of joints) it denotes the position of each of the remaining nodes, and when i is initial it denotes the initial position when the The relative coordinate position of the joint in the relative coordinate system with its parent node as the origin; denote the translation component of the node i from the current coordinate system to the coordinate system where the parent node is located at the moment t by M_i^t ($i = \text{Root}$ or $1 \le i \le \text{num}$); denote the translation component of the node i from the current coordinate system to the coordinate system where the parent node is located at the moment t by R_i^t ($i =$ Root or $1 \le i \le \text{num}$) denotes the rotation vector of the joint i around its parent joint at the moment t, which consists of components in x, y, z directions respectively.

From the perspective of hierarchical description, the position of each joint point under the global coordinate system can be derived from Eq. (17):

$$P_i^t\left(x, y, z\right) = M_{Root}^t R_{Root}^t M_1^t R_1^t \ldots M_i^t R_i^t P_{initial}^t \left(x, y, z\right) \tag{17}$$

For each frame in the motion sequence, the human body model is i.e. localized by the location of its joints. The position of the root node can be calculated from the translation and rotation of the root node, and then the position of each child node can be found step by step as shown in the following equation.

$$P_{Root}^t\left(x, y, z\right) = M_{Root}^t R_{Root}^t P_{Root}^{t-1}\left(x, y, z\right) \tag{18}$$

$$P_i^t\left(x, y, z\right) = M_i^t R_i^t P_i^{t-1}\left(x, y, z\right), \quad i = 1, 2, \ldots, num \tag{19}$$

Thus the avatar motion can be represented as a collection of translations of the root node and rotation vectors of each node at each frame, i.e:

$$Q(t)=\left(M_{Root}^{t},R_{Root}^{t},R_{i}^{t}\right),\quad i=1,2,\ldots,num \tag{20}$$

From the user's point of view, it is more intuitive and convenient to use the traditional Euler angles to define the orientation of an object. However, interpolation is often required in the keyframe method of virtual human motion. Since the three angles of Euler angles do not have mutual constraints, and an equal amount of changes in Euler angles may not necessarily cause an equal amount of rotational changes, direct interpolation of Euler angles will lead to pathological phenomena, which may result in the existence of more than one Euler angle that can be described by a given orientation, i.e., the problem of "cardinal lock". Moreover, while the traditional Eulerian angle method has to rotate around each of the three axes, i.e., it requires multiple matrices to be multiplied together, quaternions are more convenient than Eulerian angles to represent the rotation of an object, and there is no redundant information. This makes it a trend for representing motion.

The conversion between quaternions and Euler angles is easy and can be found by applying the concept of hypercomplex mapping. For the $Z-Y-X$ transformation, the conversion from Euler angles to quaternions can be expressed as:

$$q=\left(\cos\frac{\psi}{2}+\sin\frac{\psi}{2}\cdot k\right)\circ\left(\cos\frac{\theta}{2}+\sin\frac{\theta}{2}\cdot j\right)\circ\left(\cos\frac{\phi}{2}+\sin\frac{\phi}{2}\cdot i\right) \tag{21}$$

Expanding (21) yields the Euler angle to quaternion conversion formula:

$$\begin{bmatrix}q_0\\q_1\\q_2\\q_3\end{bmatrix}=\begin{bmatrix}\cos\frac{\psi}{2}\sin\frac{\phi}{2}\cos\frac{\theta}{2}-\sin\frac{\psi}{2}\cos\frac{\phi}{2}\sin\frac{\theta}{2}\\\sin\frac{\psi}{2}\sin\frac{\phi}{2}\cos\frac{\theta}{2}+\cos\frac{\psi}{2}\cos\frac{\phi}{2}\sin\frac{\theta}{2}\\-\cos\frac{\psi}{2}\sin\frac{\phi}{2}\sin\frac{\theta}{2}+\sin\frac{\psi}{2}\cos\frac{\phi}{2}\sin\frac{\theta}{2}\\\sin\frac{\psi}{2}\sin\frac{\phi}{2}\sin\frac{\theta}{2}+\cos\frac{\psi}{2}\cos\frac{\phi}{2}\sin\frac{\theta}{2}\end{bmatrix} \tag{22}$$

Based on the above virtual man driving algorithm, the virtual man motion simulation method is as follows:

Step 1: Read in the motion capture data;

Step 2: Determine the structure of the virtual man from the joints of the read-in motion capture data;

Step 3: Establish the virtual human model according to the structure of the virtual human;

Step 4: Determine the position of each joint point in the current frame from the motion capture data;

Step 5: Calculate the angle and relative position of the corresponding joints;

Step 6: Unify the data of each joint point into the global coordinate system;

Step 7: display the current position;

Step 8, the pose of this frame is completed, the loop jumps to step 4 and starts the next frame.

3. Experimental results

3.1. Experimental platforms

All subsequent research work is based on an experimental platform that consists of a physical humanoid robot, a virtual environment for that robot, and experimental data collected based on that robot. In order to avoid repeating the experimental environments in the subsequent chapters, the experimental environments involved are described here in a unified manner. The experimental environment is introduced here in terms of the robot platform and the experimental dataset, respectively. The experimental platform adopts the AlphalP robot as the experimental platform. In order to enhance the playability of the AlphalP robot, the company that produces the AlphalP robot created a dance-sharing platform where users can share their music-based choreographed dance moves with each other. There are a large number of music-based choreographed dances on its dance-sharing platform, which are excellent dance resources that can be used to create datasets, which is an important reason for choosing this robot as an experimental platform. In all the subsequent research work, the datasets are produced based on the high-quality dances with musical accompaniment collected from this platform.

3.2. Dance movement generation results

The real samples used in the model training were used to produce a dataset based on 60 high-quality dances with musical accompaniment collected from this dance sharing platform. The produced dance pose dataset contains 2900 different dance poses. Due to the limited number of samples in the set of dance poses, the model is trained with all the data once as a training cycle. The model can generate high quality dance poses after about 150 cycles of training. The change of loss value and the change of dance pose quality during the training of the network are shown in Fig. 4. During training, the performance of the network shows a trend of increasing and then decreasing. The similarity between the dance poses generated by the network and the poses in the training set stabilizes above 0.80 at the late stage of training. When the similarity of the dance poses is greater than 0.75 or more, the two are already visually similar. In order to test the performance of the TransGAN network in evaluating the dance poses, the performance of the network is examined here by comparing its evaluations of the random dance poses, the generated dance poses, and the real dance poses. First, 100 random dance poses are generated, and then 100 generated dance poses are randomly generated using TransGAN network. Afterwards these two test sets and the real dance pose dataset were sequentially fed into the TransGAN network in order to test the TransGAN network's evaluation of each of them. Since the positive samples are labeled 1 and the negative samples are labeled -1 in the loss function setting of the TransGAN network, the output of the TransGAN network is expected to be closer to 1 the more the data matches the distribution of the real dance poses, and vice versa closer to -1. The results of the experiments on the ratings of the different sources of the dances are shown in Fig. 5, where the output values of the real dance poses and the generated poses ranged from -0.4 to -0.7 Between them, the ratings of the two can no longer be completely separated, while the output values of the randomly generated dances are all between -1.0 and -1.8, which are completely separable from the other two types of dances, a feature that allows the TransGAN network to be used in the problem of classification of dance styles or genres.

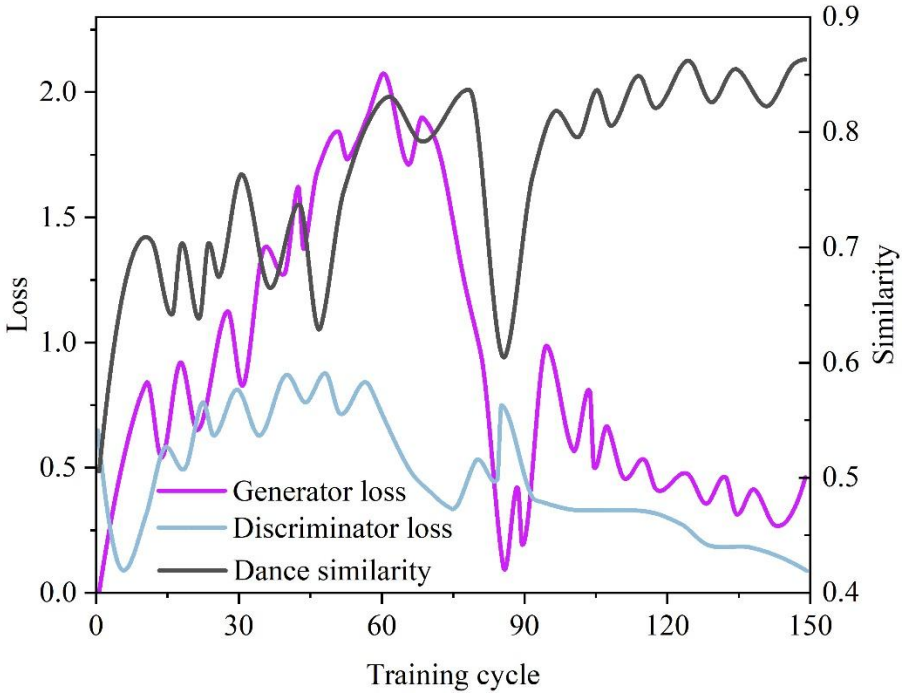

Figure 4. Changes in loss values and changes in the quality of the dance.

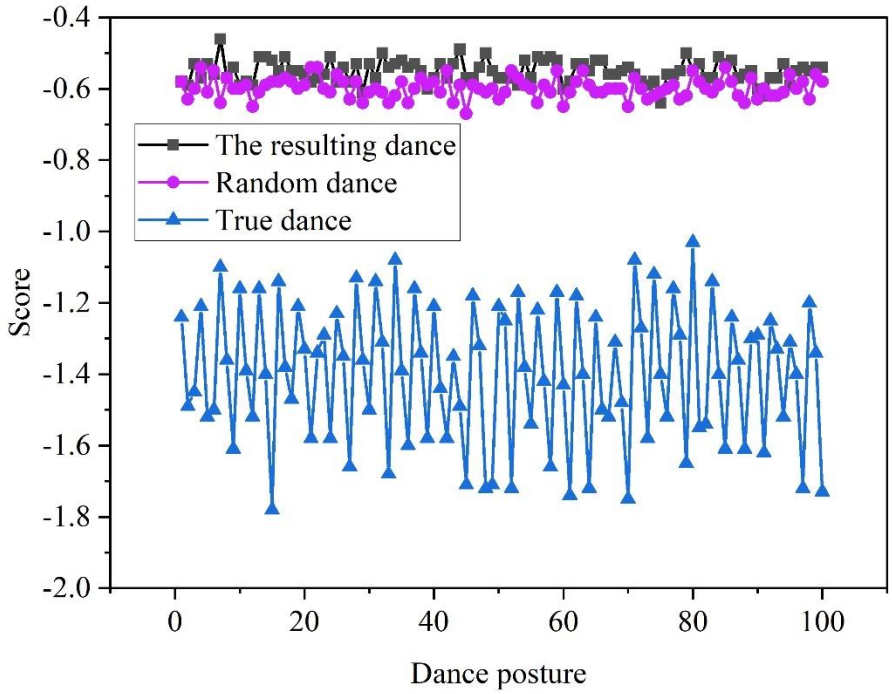

Figure 5. The discriminator scores the different sources of dance.

Firstly, a music file is sliced according to the window length (100 frames or 160 frames), and the underlying features are extracted for each section of music; then the i-1th frame of the model outputs the action features and the i-frame of the music underlying features are input to get the i-frame of the model outputs the action features (in which the input action features of the 0th frame are the given real action features). Each frame is generated iteratively in a loop according to the window length, and then each frame of action is connected to get a piece of dance action with the same length as the window. Table 1 shows the evaluation results of the TransGAN model in a window of 100 frames. Bolded numbers indicate that the TransGAN model achieves better results than the CL-M2M model, and italicized bolded numbers indicate that the evaluation values of the two models are equal. Where HR(90) and HR(170) denote the hit rate for clustering numbers of 90 and 170, GS denotes the style score, and OS(90) and OS(170) denote the overall score for clustering numbers of 90 and 170, respectively. From the table, it can be seen that the TransGAN model outperforms the CL-M2M model in both OS (90) and OS (170) overall scores for the six categories of Korean, Viennese, Tibetan, Cyprus, Hiphop and Salsa. There were also six categories in HR (90) that were greater than or equal to the CL-M2M model score, and eight categories in HR (180) that were greater than or equal to the CL-M2M model score. On the style score GS the TransGAN model scores are better than or equal to the CL-M2M model scores except for the dimensionality family.

Table 1. The average assessment of the model in 100 frames.

Style	HR (90)	HR (170)	GS	OS (90)	OS (170)
Korean	**0.40**	**0.24**	**0.98**	**0.65**	**0.62**
Dai	0.14	**0.06**	**0.99**	0.54	**0.53**
Classical Dance	0.15	***0.08***	**1.00**	**0.58**	0.56
Uighur	**0.42**	**0.33**	0.95	**0.72**	**0.68**
Tibetans	**0.38**	0.15	**0.98**	**0.63**	**0.59**
Cyprus	**0.32**	**0.35**	***1.00***	**0.68**	**0.63**
Groovenet	**0.19**	**0.16**	***1.00***	0.54	**0.54**
Hiphop	**0.13**	**0.14**	**0.05**	**0.14**	**0.13**

Salsa	**0.35**	**0.25**	**0.94**	**0.66**	**0.67**

Table 2 shows the evaluation results of the TransGAN model in the 160-frame window. Similarly, HR (90) and HR (170) denote the hit rate for clustering numbers of 90 and 170, respectively, GS denotes the style score, and OS (90) and OS (170) denote the overall score for clustering numbers of 90 and 170, respectively. Among them, HR (90) has 7 categories greater than or equal to the CL-M2M model score, HR (175) has 6 categories better than the CL-M2M model, and GS, OS (90) and OS (175) all have 5 categories greater than or equal to the CL-M2M model. It shows that most of the time the TransGAN model achieved optimal results and outperformed the CL-M2M model.

Table 2. Average assessment of the 160 frame window.

Style	HR (90)	HR (175)	GS	OS (90)	OS (175)
Korean	**0.45**	0.31	0.92	***0.75***	0.68
Dai	0.08	**0.14**	0.75	0.46	**0.44**
Classical Dance	0.15	0.08	***1.00***	0.57	0.54
Uighur	**0.37**	**0.15**	0.97	**0.69**	0.53
Tibetans	**0.58**	**0.27**	**0.99**	**0.77**	**0.62**
Cyprus	**0.35**	**0.27**	***1.00***	**0.67**	**0.65**
Groovenet	**0.31**	**0.25**	***1.00***	0.66	**0.67**
Hiphop	**0.18**	**0.12**	**0.97**	**0.59**	**0.56**
Salsa	**0.24**	0.18	0.98	0.63	0.59

GrooveNet does not publicize its models and code, but on their homepage they have action data they generated on randomly selected music files. In this experiment, 2 of these music files were selected and the results were generated on TransGAN, CL-M2M and GrooveNet models. As Figure 6 and Figure 7 show the comparison of the generation results of the four models on Dance 1 and Dance 2, respectively, it can be seen that GrooveNet does not generate effective dance movements on the test data, and each frame of its generation results does not conform to the physiological significance of the human body.

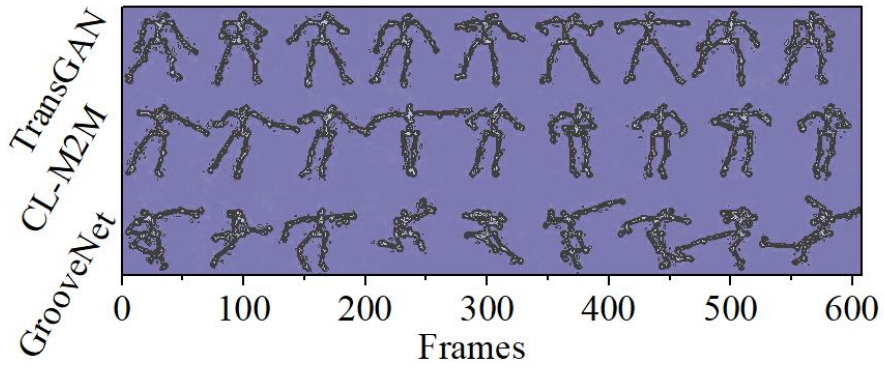

Figure 6. The resulting results of the model on dance 1.

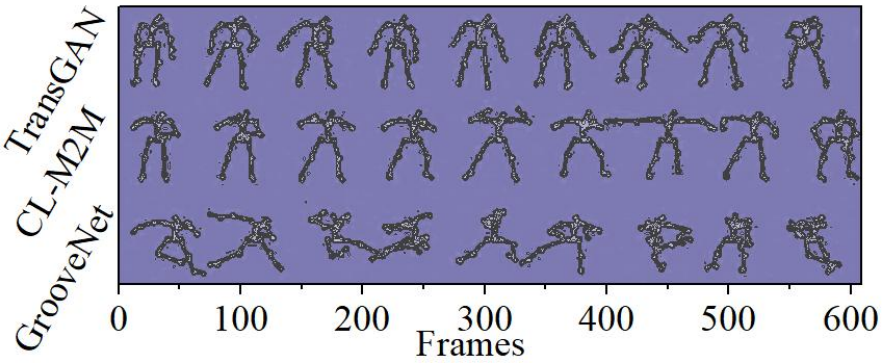

Figure 7. The resulting results of the model on dance 2.

Figures 8 and 9 compare the hand speeds of the two action sequences generated by the TransGAN and CL-M2M models, where the horizontal axis indicates the number of frames in which the action is located and the vertical axis indicates the magnitude of the hand speed. From the figure, it can be seen that the results of the CL-M2M model show sudden velocity changes, and both dance actions 1 and 2 show more than 5 velocity changes, while the dance action sequences generated by the TransGAN model are smoother.

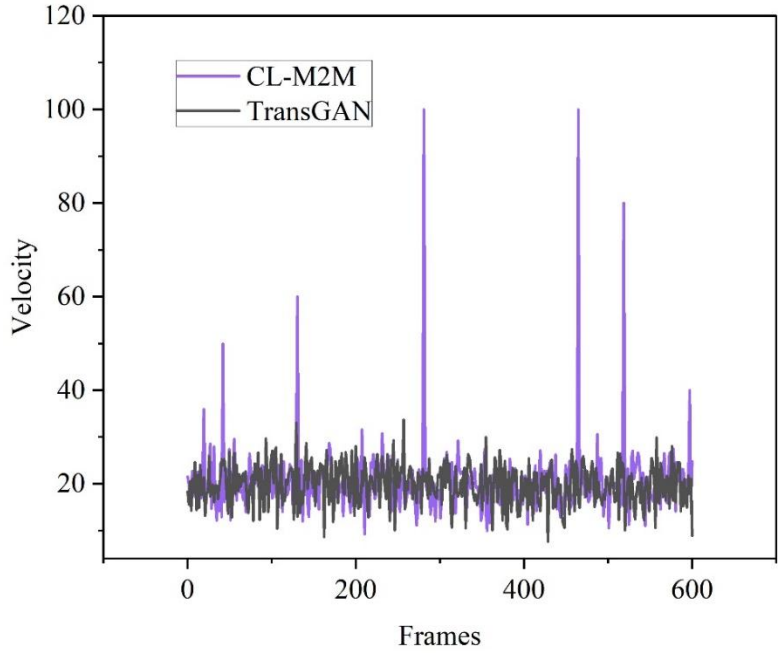

Figure 8. The hand speed of the sequence of action 1.

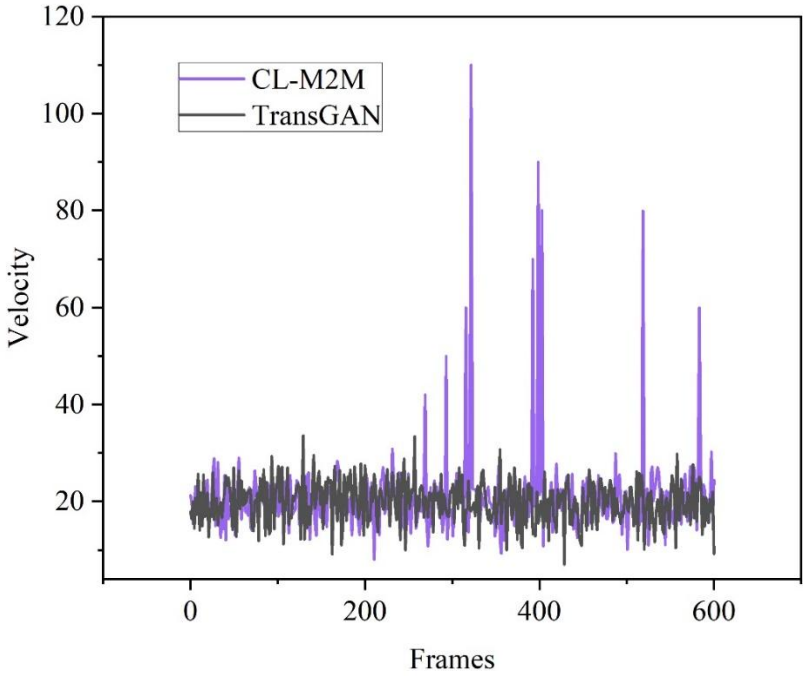

Figure 9. The hand speed of the sequence of action 1.

3.3. Virtual human simulation experiment results

3.3.1. Virtual human stability motion simulation experiment and analysis

Human dynamic motion is inherently stable and resilient to moderate perturbations. A virtual human motion controller is not only to ensure the generation of human motion that looks like the real one, but also to maintain balance while remaining robust to perturbations. In the simulation experiments, walking motion data extracted from the video is input to simulate the balance recovery ability of the virtual human. The process of recovering the balance of the virtual human after receiving a push from an external force during walking is simulated. The controller is simulated in test mode for 15 seconds at a time, and a 500N external push force lasting for 1 second is applied during the simulation. After several statistical analyses, it can be found that most of the peaks of lateral displacements occur in the first three steps after the perturbation, and the equilibrium is restored within three steps, and further deviation is stopped and can return to the reference line. In a few anomalous cases, the avatar failed to walk continuously after regaining equilibrium, or failed to return to the reference line. In addition, the performance of the avatar based on motion capture data and the avatar with only skeletal structure under external force pushing was compared in the simulation platform. The experimental results showed that the virtual human based on motion capture data did not lose balance and fall after applying a push force of 500N. However, the virtual human with a skeletal model was unable to maintain balance after applying an external force. Next, further simulation experiments were conducted on the virtual human based on motion capture data by increasing the thrust force by 15N each time, continuously applying the external force and observing the performance of the virtual human until the thrust force reached about 680N and the virtual human fell. The results show that the virtual human model based on motion capture data constructed in this study is more stable and resistant to interference than the virtual human with only skeletal structure.

3.3.2. Virtual human diversity motion simulation experiment and analysis

In order to validate the effectiveness of the virtual human diverse motion simulation system constructed in this study, eight sets of single human motion videos (including walking, running, dancing, open and close jumping, frog jumping, single leg rotation, single leg squatting and kicking with different difficulties) filmed outdoors were inputted into the motion simulation system. The virtual human was able to simulate a variety of human motions in the simulation environment, exhibiting natural, smooth and realistic characteristics similar to real human movements.

In order to evaluate the performance of the virtual human diverse motion simulation system constructed in this study, the virtual human motion simulation results are compared with the joint-driven method using only the motion tracking layer to verify the accuracy of the virtual human simulation in

generating diverse motions in this paper, and the joint errors of different motions are shown in Table 3.

The ability of the virtual human based on motion capture data to mimic the input reference motion is superior to the virtual human motion using only the joint drive. Regardless of the difficulty of the input motion, the average error of the motion capture data-based joints is smaller than that of the joint-driven virtual human motion, and it can achieve highly accurate imitation in relatively simple walking and running motions, and the error of the hip joint of the walking motion is only 1.006, which indicates that the motion capture data-driven virtual human model can more accurately mimic the reference motions, and it has better motion control ability.

Table 3. The joint error of different movements.

Sports	Arthrosis	Algorithm	Mean error	Sports	Arthrosis	Algorithm	Mean error
Walk	Hip joint	Joint-Driven	1.125	Run	Hip joint	Joint-Driven	2.449
		Motor data-Driven	1.006			Motor data-Driven	1.341
	Knee joint	Joint-Driven	2.342		Knee joint	Joint-Driven	3.218
		Motor data-Driven	1.184			Motor data-Driven	1.451
	Shoulder joint	Joint-Driven	4.239		Shoulder joint	Joint-Driven	4.896
		Motor data-Driven	3.326			Motor data-Driven	4.233
	Elbow joint	Joint-Driven	3.348		Elbow joint	Joint-Driven	3.983
		Motor data-Driven	2.854			Motor data-Driven	3.26
Open jump	Hip joint	Joint-Driven	8.676	Kick	Hip joint	Joint-Driven	5.217
		Motor data-Driven	6.333			Motor data-Driven	2.41
	Knee joint	Joint-Driven	4.884		Knee joint	Joint-Driven	7.689
		Motor data-Driven	3.234			Motor data-Driven	2.172
	Shoulder joint	Joint-Driven	5.904		Shoulder joint	Joint-Driven	6.217
		Motor data-Driven	2.677			Motor data-Driven	4.332
	Elbow joint	Joint-Driven	2.538		Elbow joint	Joint-Driven	8.682
		Motor data-Driven	1.836			Motor data-Driven	5.683
Dance	Hip joint	Joint-Driven	5.448	Leapfrog	Hip joint	Joint-Driven	15.497
		Motor data-Driven	4.37			Motor data-Driven	12.007
	Knee joint	Joint-Driven	7.479		Knee joint	Joint-Driven	12.759
		Motor data-Driven	3.235			Motor data-Driven	10.3

	Shoulder joint	Joint-Driven	5.887		Shoulder joint	Joint-Driven	10.496
		Motor data-Driven	2.683			Motor data-Driven	9.397
	Elbow joint	Joint-Driven	6.433		Elbow joint	Joint-Driven	8.985
		Motor data-Driven	5.703			Motor data-Driven	6.392
Squat down	Hip joint	Joint-Driven	9.445	Rotations	Hip joint	Joint-Driven	10.667
		Motor data-Driven	6.559			Motor data-Driven	8.454
	Knee joint	Joint-Driven	12.12		Knee joint	Joint-Driven	12.225
		Motor data-Driven	9.231			Motor data-Driven	10.295
	Shoulder joint	Joint-Driven	5.786		Shoulder joint	Joint-Driven	8.458
		Motor data-Driven	4.948			Motor data-Driven	5.415
	Elbow joint	Joint-Driven	6.892		Elbow joint	Joint-Driven	11.769
		Motor data-Driven	6.441			Motor data-Driven	9.429

3.4. Kinematic performance analysis in rugged terrain

The terrain is usually set as an ideal horizontal surface in simulation experiments, but the terrain in the real environment has a certain complexity, in order to further prove the adaptability of the algorithm and the application value of migrating the algorithm to the physical robot, the simulation ground is set as an uneven uneven and convex terrain, and the walking state of the virtual man is observed. Figure 10 shows the height changes of the virtual human head, shoulder, center of mass, knee and ankle.

From the figure, it can be seen that the virtual man in the face of concave and convex terrain with jagged height difference, the stability of the virtual man is significantly reduced, and the walking process is more fluctuating, especially in the region of the terrain height transition, it is necessary to maintain the balance of the posture after a period of time of gait self-adjustment, but it is still in the uneven terrain with the ability of complete walking, showing a better adaptability.

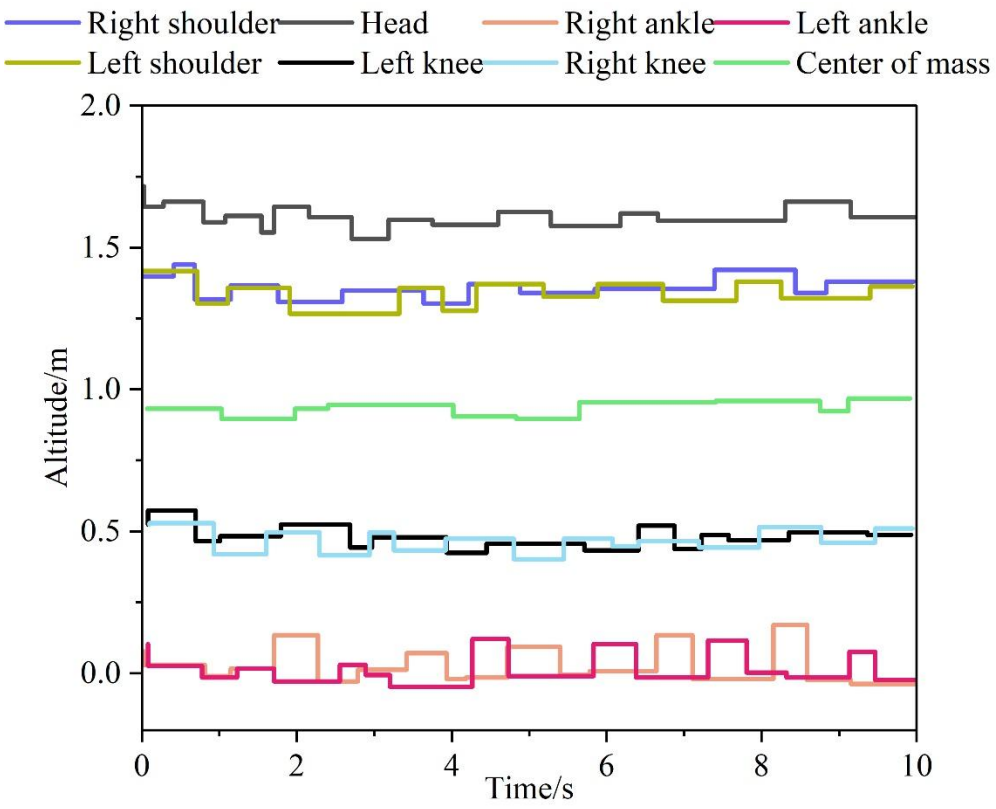

Figure 10. The convex topography of the convex topography is highly varied.

4. Conclusion

This study aims to solve the technical bottleneck of automated dance movement creation by constructing a framework for dance movement generation and virtual dancer driving based on TransGAN generative model.

The results of the dance movement generation experiments show that the posture similarity of the network-generated dance movements in this paper is stabilized above 0.80 in the late stage, and the output values of the real dance movements and the generated dance movements are between -0.4 and -0.7, and the difference in the scores between the two is small, and the difference in the scores between the two and the randomly-generated dance movements is obvious. Compared with the CL-M2M model, GrooveNet and other similar models, the model in this paper generates more reasonable dance movements and smoother dance movement sequences. The virtual simulation experiments further confirm that the generated data can drive the virtual dancer characters, showing good performance in terms of stability and movement diversity, and good adaptability on non-flat terrain.

This paper provides an intelligent tool for digital dance content creation. The potential of combining Transformer and GAN for the task of continuous dance sequence generation is explored. The technique can be applied in neighborhoods such as game development and dance teaching, which greatly improves the efficiency of creators.

This study also has some limitations; the stylistic generalization ability of the model-generated movements needs to be further improved, and the system has not yet been involved in the generation of fine movements such as finger and facial expressions. Future research will be devoted to the introduction of cross-stylistic transfer learning and the integration of emotion labeling for dance movement generation.

References

1. Peng, X. (2024). Historical development and cross-cultural influence of dance creation: Evolution of body language. Herança, 7(1), 88-99.
2. Yazaki, Y., Soga, A., Umino, B., & Hirayama, M. (2015, October). Automatic composition by body-part motion synthesis for supporting dance creation. In 2015 International Conference on Cyberworlds (CW) (pp. 200-203). IEEE.
3. Gao, L. X. (2024). Dance Creation Based on the Development and Application of a Computer Three-Dimensional Auxiliary System. International Journal of Maritime Engineering, 1(1), 347-358.
4. Peng, Z. (2023). The Current Situation and Development of Dance Creation in the New Media Era. Art and Performance Letters, 4(5), 32-38.
5. Shi, J. (2021, October). Application of 3d computer aided system in dance creation and learning. In International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy (pp. 88-95). Cham: Springer International Publishing.
6. Ruthotto, L., & Haber, E. (2021). An introduction to deep generative modeling. GAMM-Mitteilungen, 44(2), e202100008.
7. Shaham, T. R., Dekel, T., & Michaeli, T. (2019). Singan: Learning a generative model from a single natural image. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 4570-4580).
8. Tomczak, J. M. (2024). Why deep generative modeling?. In Deep generative modeling (pp. 1-13). Cham: Springer International Publishing.
9. Song, Y., & Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems, 32.
10. Zeng, D. (2025). AI-Powered Choreography Using a Multilayer Perceptron Model for Music-Driven Dance Generation. Informatica, 49(20).
11. Ferreira, J. P., Coutinho, T. M., Gomes, T. L., Neto, J. F., Azevedo, R., Martins, R., & Nascimento, E. R. (2021). Learning to dance: A graph convolutional adversarial network to generate realistic dance motions from audio. Computers & Graphics, 94, 11-21.
12. Huang, J., Huang, X., Yang, L., & Tao, Z. (2024). D2MNet for music generation joint driven by facial expressions and dance movements. Array, 22, 100348.
13. Jin, Y. (2022). A Three-Dimensional Animation Character Dance Movement Model Based on the Edge Distance Random Matrix. Mathematical Problems in Engineering, 2022(1), 3212308.
14. Wallace, B., Hilton, C., Nymoen, K., Torresen, J., Martin, C. P., & Fiebrink, R. (2023, June). Embodying an interactive AI for dance through movement ideation. In Proceedings of the 15th Conference on Creativity and Cognition (pp. 454-464).
15. Lu, Q. (2025). Digital dance generation and application based on hybrid density network. International Journal of Information and Communication Technology, 26(2), 51-66.
16. Wallace, B., Martin, C. P., Tørresen, J., & Nymoen, K. (2021, June). Learning embodied sound-motion mappings: Evaluating AI-generated dance improvisation. In Proceedings of the 13th Conference on Creativity and Cognition (pp. 1-9).
17. Zhou, J., Weber, R., Wen, E., & Lottridge, D. (2025, March). Real-Time Full-body Interaction with AI Dance Models: Responsiveness to Contemporary Dance. In Proceedings of the 30th International Conference on Intelligent User Interfaces (pp. 1177-1187).
18. Valle-Pérez, G., Henter, G. E., Beskow, J., Holzapfel, A., Oudeyer, P. Y., & Alexanderson, S. (2021). Transflower: probabilistic autoregressive dance generation with multimodal attention. ACM Transactions on Graphics (TOG), 40(6), 1-14.
19. Ma, Q. (2025). Harnessing generative neural networks to fuse traditional Tujia Baishou dance with contemporary choreography: Enhancing creativity and aesthetic experience in dance students. Acta Psychologica, 258, 105178.
20. Peng, Y. (2025). Designing AI-Driven Dance Choreography: Motion Capture and Generation Protocols. Journal of Criminal Investigation and Criminology, 76(2).
21. Strutt, D., & Cisneros, R. (2021). Virtual relationships: the dancer and the avatar. Theatre and Performance Design, 7(1-2), 61-81.
22. Kuzmin, A. I., Semyonov, D. A., & Samsonovich, A. V. (2021, September). Classification and Generation of Virtual Dancer Social Behaviors Based on Deep Learning in a Simple Virtual Environment Paradigm. In Biologically Inspired Cognitive Architectures Meeting (pp. 231-242). Cham: Springer International Publishing.
23. Ren, Y. (2025, May). Virtual Human Dance Movement Generation Technology Based on Transformer and VITON Network. In 2025 2nd International Conference on Intelligent Computing and Robotics (ICICR) (pp. 454-458). IEEE.
24. Liu, X., & Ko, Y. C. (2022). The use of deep learning technology in dance movement generation. Frontiers in Neurorobotics, 16, 911469.
25. Kritsis, K., Gkiokas, A., Pikrakis, A., & Katsouros, V. (2022). Danceconv: Dance motion generation with convolutional networks. Ieee Access, 10, 44982-45000.
26. Zhang, C., Tang, Y., Zhang, N., Lin, R. S., Han, M., Xiao, J., & Wang, S. (2024). Bidirectional autoregessive diffusion model for dance generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 687-696).
27. Jiang, H., & Yan, Y. (2024). Sensor based dance coherent action generation model using deep learning framework. Scalable Computing: Practice and Experience, 25(2), 1073-1090.

28. Ma, X., & Wang, K. (2022). Dance action generation model based on recurrent neural network. Mathematical Problems in Engineering, 2022(1), 8455961.
29. Cai, X., Xi, M., Jia, S., Xu, X., Wu, Y., & Sun, H. (2023). An automatic music-driven folk dance movements generation method based on sequence-to-sequence network. International journal of pattern recognition and artificial intelligence, 37(05), 2358003.
30. Gou, J. (2025, July). Research on deep generation model for automatic dance movement generation and style conversion. In IET Conference Proceedings CP941 (Vol. 2025, No. 25, pp. 371-375). Stevenage, UK: The Institution of Engineering and Technology.
31. Wu, Y., Wu, Z., & Ji, C. (2025). Transformer-based partner dance motion generation. Engineering Applications of Artificial Intelligence, 139, 109610.
32. Yue Fan, Shumiao Chen,Jia Feng, Yongpeng Shi, Yue Pan, Rende Ma & Mingsheng Niu. (2025). A variable channels multi-pass cell TDLAS-based trace gas sensor with convolution neural network and empirical modal decomposition algorithm. Optics and Laser Technology,192(PC),113700-113700.https://doi.org/10.1016/J.OPTLASTEC.2025.113700.
33. Xiaowen Sun, Jiahao Li, Guiying Yan & Renmin Han. (2025). ADMGCN: Graph Convolutional Network for Alzheimer's Disease Diagnosis with a Meta-learning Paradigm.. Bioinformatics (Oxford, England),https://doi.org/10.1093/BIOINFORMATICS/BTAF580.
34. Weitao Gao, Chenbin Wang, Changxin Chen, Tiehua Ma & Wenchao Guo. (2025). Spatio-Temporal Feature Extraction for Human Action Recognition in Visual Communication Systems Using Improved ST-GCN. Journal of Circuits, Systems and Computers,34(11),https://doi.org/10.1142/S0218126625502391.
35. Mashael M. Asiri, Kholoud Alghamdi, Fahad Alzahrani & Mahir Mohammed Sharif. (2025). An innovative multi-head attention mechanism-driven recurrent neural network model with feature representation fusion for enhanced image captioning to assist individuals with visual impairments. Scientific Reports,15(1),35845-35845.https://doi.org/10.1038/S41598-025-19733-W.