



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642

<https://www.inderscience.com/ijict>

Student dance teaching motion recognition and classification based on spatial temporal graph convolutional networks

Lin Ma

DOI: [10.1504/IJICT.2026.10076627](https://doi.org/10.1504/IJICT.2026.10076627)

Article History:

Received:	07 November 2025
Last revised:	10 December 2025
Accepted:	11 December 2025
Published online:	11 March 2026

Student dance teaching motion recognition and classification based on spatial temporal graph convolutional networks

Lin Ma

School of Music and Dance,
ABA Teachers College,
Ngapa, 623000, China
Email: lin6687789@sina.com

Abstract: This study proposes a method specifically designed for recognising students' dance teaching movements based on spatial temporal graph convolutional networks. The method employs three parallel spatial temporal aggregation graph convolutions to extract dynamic features. By incorporating OpenPose for human pose estimation, the proposed spatial temporal graph convolutional network-based action recognition method effectively classifies student dance teaching movements. Experimental model achieving recognition accuracies of 89.1% and 90.4% on cross-subject and cross-set benchmarks respectively, compared with existing advanced models, the improvement ranges from 0.3 to 4.0 percentage points and 0.3 to 3.6 percentage points respectively. Furthermore, evaluation of the self-constructed dataset encompassing ten classical dance categories demonstrates that the method achieves recognition accuracy ranging from 89.6% to 96.1%, effectively resolving the issue of inaccurate dance movement recognition.

Keywords: spatial temporal graph convolution network; dance movement recognition; human pose estimation.

Reference to this paper should be made as follows: Ma, L. (2026) 'Student dance teaching motion recognition and classification based on spatial temporal graph convolutional networks', *Int. J. Information and Communication Technology*, Vol. 27, No. 18, pp.44–59.

Biographical notes: Lin Ma is an Associate Professor in the School of Music and Dance at ABA Teachers College. She received a Bachelor's degree in Choreography from Shandong Youth University of Political Science in 2007. Her research interests include dance education, choreography, ethnic folk dance, and dance movement recognition.

1 Introduction

With the rapid advancement of artificial intelligence, computer vision applications have experienced unprecedented growth across diverse domains including healthcare (Lai et al., 2025), human-machine interaction (Al-Faris et al., 2020), sports analysis (Ding et al., 2022; Wu et al., 2023), and education (Dimitriadou and Lanitis, 2024; Pareek and Thakkar, 2020). The integration of intelligent systems into educational

frameworks has opened new avenues for personalised learning experiences, automated assessment, and real-time feedback mechanisms (Alam and Mohanty, 2023; George and Wooden, 2023; Kuchynska et al., 2025). Among these applications, action recognition has emerged as a fundamental task that enables machines to understand and interpret human behaviours through visual analysis, providing significant potential for enhancing educational methodologies (Mao, 2021).

In the context of dance education, traditional teaching methods heavily rely on manual observation and subjective assessment by instructors (Qin and Meng, 2025), which presents several inherent limitations (Guo et al., 2021). Students often struggle to receive immediate feedback on their performance, making it challenging to identify and correct movement errors in real-time. Moreover, the subjective nature of human evaluation can lead to inconsistencies in assessment criteria, particularly when multiple instructors are involved. The growing demand for standardised dance education, coupled with the increasing adoption of digital learning platforms, has created an urgent need for technology-supported innovative teaching and assessment methods (Pastukhov, 2021).

Current human motion recognition methods have advanced rapidly through the development of sophisticated deep learning architectures (Bian et al., 2021; Wang et al., 2013; Yang et al., 2022). These challenges stem from the high complexity of dance movements, self-occlusion of the human body during performance, the inherent fine-grained long-term temporal dependencies within dance sequences, and the subtle yet critical distinctions required to differentiate dance styles (Wang and Jan, 2022). Traditional convolutional neural networks (CNNs), though highly effective for image-based tasks, are limited in capturing the graph-structured topology of human skeletal data. Recurrent neural networks (RNNs), including variants such as LSTMs, are capable of handling temporal sequences but often fall short in modelling the high-d al relationships and fine-grained motion patterns present in dance movements. These limitations have impeded progress toward building accurate, robust, and scalable dance recognition systems, particularly those intended for educational applications where precision and explainability are crucial. There is a growing need for novel frameworks that can integrate spatial and temporal features more effectively to address the unique demands of dance movement analysis.

In recent years, graph convolutional networks have demonstrated significant potential in addressing the aforementioned challenges (Zhang and Li, 2022). By representing human posture as a graph structure, wherein joints serve as nodes and bones as edges, these networks effectively model the spatial and temporal dependencies inherent in human motion data (Wang et al., 2024). However, existing graph-based approaches remain limited in capturing the dynamic characteristics of dance movements, particularly the intricate interactions between spatial joint relationships and temporal motion patterns that define distinct dance styles. The emergence of spatial temporal modelling techniques offers a potential solution to these issues. The spatial temporal graph convolutional networks (ST-GCN) model enhances human action recognition by learning spatial and temporal patterns from data, thereby achieving greater expressive power and stronger generalisation capabilities. (Wang et al., 2022; Zhang et al., 2021) By simultaneously considering spatial and temporal information, these methods capture more comprehensive motion representations.

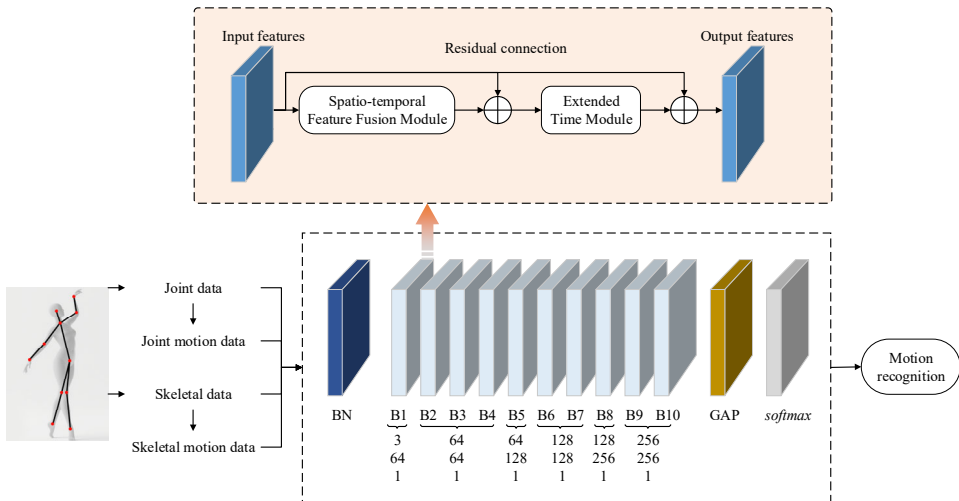
In summary, to address the aforementioned challenges in automated dance instruction, this study proposes an extended temporal-spatial feature fusion graph convolutional network (TSFF-GCN) specifically tailored for student dance motion

recognition and classification, based on ST-GCN. The innovation of this research manifests in two aspects: firstly, the development of a novel spatial temporal feature fusion module that employs parallel graph convolutions to extract dynamic spatial temporal features, while integrating an attention mechanism to achieve adaptive feature fusion; secondly, the introduction of an extended temporal module that utilises multi-scale dilated convolutions combined with a bottleneck design to capture temporal patterns at different granularities while maintaining computational efficiency.

2 Extended temporal and spatial temporal feature fusion graph convolutional network for skeleton-based human action recognition

To accurately identify students' dance teaching movements, this study proposes an extended temporal and spatial temporal feature fusion graph convolutional network. Its architecture is depicted in Figure 1, with input data dimensions of $C_{in} \times T \times N$, where C_{in} denotes the number of input channels, T represents the number of frames per input sequence, and N indicates the number of human body joints contained within the sequence. Following batch normalisation, the input data undergoes training through ten basic blocks to extract spatial temporal feature maps. Each basic block comprises a spatial temporal feature fusion module and an extended temporal module. Below each basic block, three numerical values are displayed. Taking the values beneath B1 as an example, these denote: B1 has 3 input channels, 64 output channels, and a stride of 1 for its temporal convolutional layer. To achieve dimensionality reduction, the stride of the temporal graph convolutional layer is set to 2 in the fifth and eighth basic blocks when the output channel count doubles. Spatial temporal feature maps are resized uniformly via a global average pooling layer and undergo behavioural classification through the normalised exponential function Softmax.

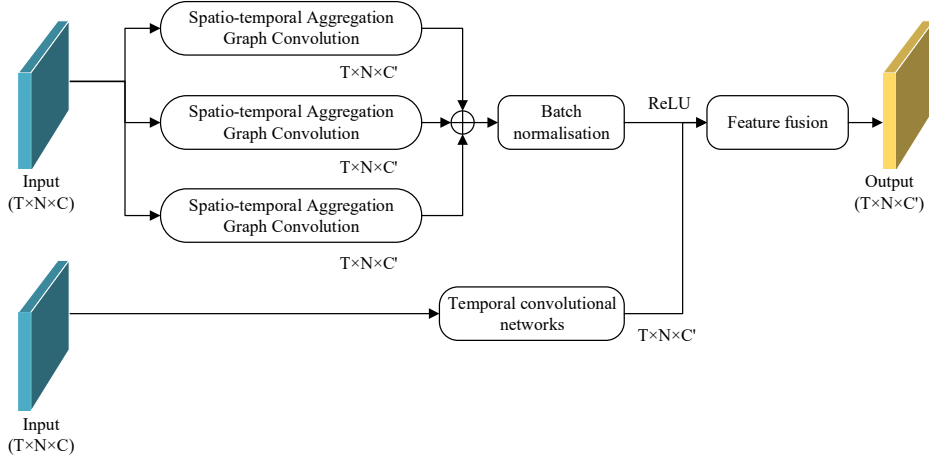
Figure 1 TSFF-GCN's overall and basic block structure (see online version for colours)



2.1 Spatial temporal feature fusion module

Figure 2 depicts the structure of the spatial-temporal feature fusion module. This module employs a tri-parallel spatial temporal aggregation graph convolution architecture to extract dynamic spatial temporal features, utilising a temporal convolutional network to extract temporal features. Finally, an attention-based feature fusion module dynamically integrates the temporal features F_{TCN} and dynamic spatial temporal features F_{STC} .

Figure 2 Structure of the spatiotemporal aggregation graph convolution submodule (see online version for colours)



The spatial temporal aggregation graph convolutional submodule employs a dual-branch architecture to extract spatial temporal features. The first branch employs a channel topology modelling approach to learn channel topologies with greater precision. While learning shared prior topologies, this branch generates distinct channel correlations based on varying inputs, thereby enabling dynamic acquisition of spatial topologies. Specifically, the input feature x undergoes dimensionality reduction via a 2D convolution with a reduction ratio of r_1 followed by a temporal pooling layer, as formulated below:

$$x' = f_{temporal}(f_{conv}(x)) \quad (1)$$

Secondly, the channel correlation $Q \in R^{N \times N \times C'}$ is obtained using the channel correlation modelling function $M(\cdot)$. For any two given nodes v_i and v_j , the formula for the channel correlation modelling function is as follows:

$$q_{ij} = M(x'_i, x'_j) = f_{conv}(\sigma\mu(x'_i, x'_j)), \quad i, j \in \{1, 2, 3, \dots, N\} \quad (2)$$

where σ is the activation function, and μ is mainly used to calculate the distance between x_i and x_j in the channel dimension. Since Q is asymmetric, $q_{ij} \neq q_{ji}$.

Finally, based on the given prior topology (the adjacency matrix $A \in R^{N \times N}$) and the topology refinement function $R(\cdot)$, the channel correlation Q is used to refine A , resulting in the refined channel topology $T \in R^{N \times N \times C'}$:

$$T = R(Q, A) = \varepsilon(A) + \alpha Q \quad (3)$$

where ε is used for standardising and dimensionally transforming the adjacency matrix A , and α is a trainable parameter used to adjust the refinement intensity.

The second branch uses a temporal convolutional network to extract temporal features, obtaining the temporal features $f_{TCN}(x)$, and combines the temporal features $f_{TCN}(x)$ with the refined channel topology T by channel aggregation. For a given channel $c_1 (c_1 \in \{1, 2, 3, \dots, C\})$, first construct a channel graph for this channel, including the refined topology T_{C_1} of c_1 and the temporal features $f_{TCN}^{C_1}$, then aggregate the two on the channel graph, and finally concatenate the outputs of all channel graphs to obtain the final output F_{out} :

$$F_{out} = CA(f_{TCN}, T) = (f_{TCN}^1 T_1) \parallel (f_{TCN}^2 T_2) \parallel \dots \parallel (f_{TCN}^C T_C) \quad (4)$$

where CA represents the channel aggregation operation, $\parallel$ indicates the concatenation operation, and the branches are concatenated in the order of channels. Each channel graph represents the relationship between nodes on the human topology graph under different motion features at different times. Since the channel topology modelling is dynamically adjusted based on different inputs, the temporal domain aggregation convolution is a dynamic graph convolution, and F_{out} is a dynamic spatiotemporal feature.

The feature fusion submodule integrates the given dynamic spatial temporal feature F_{STC} and temporal feature F_{TCN} by element-wise addition, then feeds them into a multi-scale channel attention module to extract attention weights.

The multi-scale channel attention module employs both global and local scale branches to extract attention weights. Within the global scale branch, feature maps undergo average pooling via GAP, ensuring each channel incorporates global information across the entire feature map. Subsequently, two PWConv (point-wise convolutions) facilitate interaction and fusion between channel features, thereby generating the global context. The formula for the global scale branch is expressed as:

$$G(F_{STC} + F_{TCN}) = BN \left(PWConv2 \left(\sigma \left(BN \left(PWConv1 \left(g(F_{STC} + F_{TCN}) \right) \right) \right) \right) \right) \quad (5)$$

where g denotes the global average pooling operation, BN represents batch normalisation, and σ is an activation function. The principle of the local scale branch is identical to that of the global scale. The outputs from both branches are added and activated to yield the attention weight $MS(F_{STC} + F_{TCN})$.

$$F_{STF} = M(F_{STC} \oplus F_{TCN}) \otimes F_{STC} + (1 - M(F_{STC} \oplus F_{TCN})) \otimes F_{TCN} \quad (6)$$

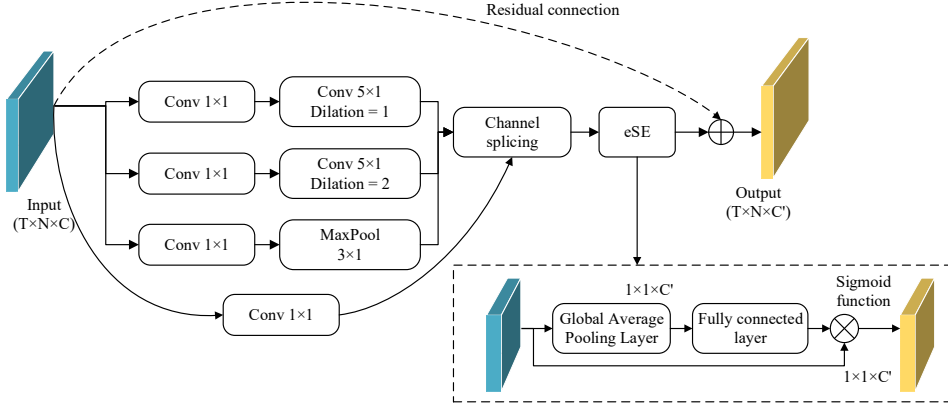
where $\oplus$ and $\otimes$ respectively denote element-wise addition and multiplication operations.

2.2 Extended time module

The extended temporal convolution module, as illustrated in Figure 3, employs a four-branch bottleneck design to reduce computational complexity and enhance efficiency. Each branch utilises a convolution kernel of size $C' \times C' / 4 \times 1 \times 1$, reducing the channel dimension to $C' / 4$. Two branches employ convolution kernels with differing expansion rates, enabling multi-temporal granularity feature extraction from skeletal data at varied sampling rates. The remaining two branches comprise a convolutional branch

and a max-pooling branch, serving to expand the model's spatial temporal receptive field. Following concatenation of the outputs from all four branches along the channel dimension, the channel dimension is elevated to C' . The resulting output is then fed into the eSE module for feature refinement.

Figure 3 Structure of the temporal expansion convolution module (see online version for colours)



The eSE module is a variant of the SE module, whose process comprises three steps: compression, excitation, and feature refinement. The compression operation employs GAP to compress the spatial temporal information of the feature map while generating channel-level global features, which can be represented by the following formula:

$$f_{GAP}(F_{MSTC}) = \frac{1}{TN} \sum_{i,j=1}^{T,N} F_{MSTC_{i,j}} \quad (7)$$

Compared to the SE module, the excitation component of the eSE module is primarily implemented through a fully connected layer with C output channels. This avoids the loss of channel information associated with using two fully connected layers for dimensionality reduction followed by dimensionality expansion. Finally, the feature refinement stage applies the obtained normalised weights to each channel via element-wise multiplication, yielding the input feature map for the eSE module. This process can be expressed as follows:

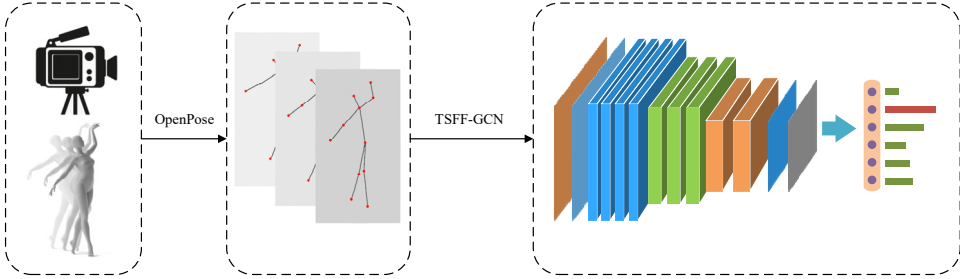
$$F_{eSE} = \delta(W_C(f_{GAP}(F_{MSTC}))) \otimes F_{MSTC} \quad (8)$$

where W_C denotes the weights of the fully connected layer, δ represents the Sigmoid function, and F_{MSTC} signifies the output of the multi-scale temporal convolution (MSTC) module. By introducing a residual connection between the input and output, the final output F_{ET} is obtained.

3 Action recognition and classification for student dance instruction based on OpenPose and TSFF-GCN

This section outlines the workflow of a behaviour recognition method based on OpenPose and ST-GCN, as depicted in Figure 4. The processing sequence comprises three principal stages: data acquisition, keypoint extraction, and feature extraction and behaviour recognition.

Figure 4 Workflow of the dance action recognition and classification method based on OpenPose and TSFF-GCN (see online version for colours)



3.1 Data collection stage

The primary task during the data acquisition phase is to furnish raw input data for the ST-GCN model employed in the feature extraction stage. Typically, graph convolutional networks require skeleton data as input, which can be obtained through two distinct methods. One approach involves direct data acquisition using Microsoft's Kinect series depth cameras. Whilst this method yields relatively precise skeleton data, it is costly and impractical. The alternative approach employs human pose estimation techniques to derive skeleton data from RGB video sequences. Whilst this method yields slightly lower accuracy than Kinect depth cameras, it offers significantly lower costs and greater practicality. Consequently, this chapter utilises RGB video sequences as input, employing OpenPose human pose estimation technology to extract skeleton data. To enhance practicality, the proposed method for human skeleton behaviour recognition based on OpenPose and graph convolutional networks employs two approaches for acquiring RGB video data: directly uploading video files to a server, or capturing video data in real-time via a camera.

To further improve the effectiveness and performance of OpenPose pose estimation, this method pre-processes the raw input video during the data acquisition stage. Specifically, YOLOv5 is employed for object detection within the input video to identify moving human targets, outputting detected human bounding boxes and confidence scores. Images are cropped based on the bounding box positional information to extract the human region, removing irrelevant background to reduce computational load on the OpenPose model and enhance the stability and accuracy of keypoint extraction.

3.2 Keypoint extraction stage

The primary task of the keypoint extraction stage is to extract skeleton data from RGB video data. This paper employs OpenPose for human keypoint extraction. OpenPose first performs denoising and edge feature enhancement on the RGB video images captured during the data acquisition stage. Subsequently, it utilises the VGG-19 backbone network for feature extraction, yielding the feature map F . Feature map F is then fed into the stage 1 module for human pose estimation. The stage 1 module comprises two branches. The first branch generates part confidence maps (PCM). This can be expressed mathematically as:

$$S^l = \rho^l(F) \quad (9)$$

where ρ' denotes the neural network within the PCM branch of the stage 1 module. PCM represents the Gaussian response of pixels at keypoints, with pixels closer to keypoints exhibiting greater response values. The second branch generates part affinity fields (PAF), expressed by the formula:

$$L^l = \phi^l(F) \quad (10)$$

where ϕ' denotes the neural network within the PAF branch of the stage 1 module. PAF is employed to represent the affinity between different keypoints, where keypoints belonging to the same individual exhibit greater affinity, while those from different individuals show lesser affinity. Given the varying detection difficulties of different keypoints and the mutual semantic information between them, OpenPose stacks multiple stage modules. The output of stage 2 is the weighted sum of the PCM, PAF, and feature map F generated by stage 1, with subsequent stages following this pattern. The PCM and PAF generated by stage t can be expressed as:

$$S^t = \rho^t(F, S^{t-1}, L^{t-1}), \quad \forall t \geq 2 \quad (11)$$

$$L^t = \phi^t(F, S^{t-1}, L^{t-1}), \quad \forall t \geq 2 \quad (12)$$

where ρ' and ϕ' denote the neural networks for the PCM branch and PAF branch respectively within the stage 1 module. At this stage, the preceding stage module can identify relatively straightforward keypoints, whilst the subsequent stage module identifies more complex keypoints based on the detection results from the prior stage. For instance, during pose estimation, certain keypoints such as the nose and eyes possess highly distinctive visual features that are readily identifiable, thus allowing detection by the earlier stage module. Conversely, other keypoints may be occluded or present greater recognition challenges. In such cases, the subsequent stage module can perform detection based on the already identified keypoints, achieving a progressively optimised effect.

To enhance prediction accuracy, OpenPose incorporates an L_2 loss function within each branch to measure discrepancies between predicted PCM/PAF and ground truth PCM/PAF. Given that certain pose estimation datasets lack complete annotations for all individuals, the loss function requires weighting. The loss functions for the PCM and PAF branches within stage t can be expressed respectively as:

$$f_S^t = \sum_{j=1}^J \sum_p W(P) \cdot \|S_j^t(p) - S_j^*(p)\|_2^2 \quad (13)$$

$$f_L^t = \sum_{c=1}^C \sum_p W(P) \cdot \|L_c^t(p) - L_c^*(p)\|_2^2 \quad (14)$$

where L_c^* denotes the ground truth PAF, the L_c^t denotes the ground truth PAF. The S_j^* denotes the true PCM. The S_j^t denotes the predicted PCM. W is a binary mask where $W(p) = 0$ when the p^{th} position in the image lacks annotation, and $W(p) = 1$ otherwise. To prevent vanishing gradients from affecting pose estimation results, gradients are periodically replenished within each stage module. The total loss for the pose estimation stage can be expressed as:

$$f = \sum_{t=1}^T f_S^t + f_L^t \quad (15)$$

After obtaining the PCM and PAF from the final stage module, OpenPose employs the principle of pair matching from graph theory to calculate affinity scores between keypoints. Using PAF correlation values as edge weights, it then applies the Hungarian algorithm to identify all bone edges belonging to the same individual. A minimum spanning tree algorithm is subsequently used to construct a skeleton model for each person, thereby achieving more precise human pose estimation.

To adapt to human behaviour detection tasks, adjustments must be made to OpenPose's relevant parameters. First, the `model_pose` parameter within the executable file must be set to COCO to reduce computational complexity. Under this configuration, OpenPose models human posture using skeleton sequences comprising only 18 keypoints. Furthermore, to minimise computational load, this task imposes a maximum detection limit of five individuals by setting the `number_people_max` parameter to 5. Consequently, for multi-person behaviours, only the five individuals with the highest confidence scores are detected.

Input images require normalisation to scale them within the range $[-1,1]$. This step enhances the model's training efficiency and convergence speed. During data acquisition, RGB video images are captured at dimensions $H \times W \times 3$, where H and W denote the image height and width respectively. Following keypoint extraction using OpenPose, the RGB image data is converted into skeleton data with dimensions equal to the *number of individuals* $\times$ *number of joints* $\times 3$. However, this skeleton data cannot be directly input into the graph convolutional network for feature extraction. Therefore, dimensionality adjustment is required, transforming the data into a tensor of batch size $\times 3 \times$ number of frames $\times$ number of joints $\times$ number of persons. The batch size is uniformly set to 1, and the number of joints is set to 18. Regarding the number of frames, a maximum limit of 300 frames is applied. Data exceeding 300 frames is processed as 300 frames, while data with 300 frames or fewer is processed according to the actual frame count.

3.3 Feature extraction and action recognition stage

The primary task of the feature extraction and action recognition stage is to utilise ST-GCN for feature extraction from skeleton data extracted by OpenPose, followed by action recognition via a Softmax classifier. This paper proposes a TSFF-GCN model based on ST-GCN for action recognition.

Due to environmental interference, occlusion, and pose variations, OpenPose frequently fails to accurately detect certain human keypoints within RGB video segments. To ensure behavioural recognition accuracy, missing keypoints within the skeleton data are first compensated for before feeding the skeleton data into the TSFF-GCN for feature extraction. The keypoints extracted by OpenPose consist of two primary elements: their two-dimensional coordinates and their corresponding confidence scores. This chapter employs historical information from the time-series data to compensate for missing keypoints, following this specific workflow:

- 1 When a frame in the video is detected to be missing the two-dimensional coordinate information of a keypoint j , trace back K frames from that frame as the coordinate origin of the time series. frames forward from this frame. Within these K frames, select the two adjacent frames where keypoint j appears. Record the coordinates (x_1, y_1) and (x_2, y_2) within these two frames. Simultaneously, trace back K frames and select the two adjacent frames containing this keypoint within the K frames, recording the coordinates (x_3, y_3) and (x_4, y_4) of keypoint j in these two frames. As human movement does not follow linear trajectories but rather curves that curve outwards from the body's centre of gravity, the motion trajectory of keypoint j prior to the current frame can be calculated using (x_1, y_1) and (x_2, y_2) . Subsequently, using (x_3, y_3) and (x_4, y_4) to compute the trajectory of keypoint j after this frame. The intersection point of these two trajectories then yields the two-dimensional coordinate information for keypoint j in this frame.
- 2 For missing confidence information of keypoint j in a given frame, trace back and forward two adjacent frames containing keypoint j using that frame as the coordinate origin. Record the confidence information and then perform a normalised weighted average processing:

$$s = \sum_{i=1}^4 \frac{S_i}{\sqrt{2\pi}} e^{-\frac{K_i^2}{2}} \quad (16)$$

where s represents the confidence information of keypoint j in the current frame, i represents the number of keypoints, s_i represents the confidence information of the four keypoints before and after, K_i represents the frame difference between the frame where the keypoint with confidence information s_i is located and the current frame, and the maximum value of K_i is 5.

- 3 Compensate for keypoints from the first frame to the last frame of the video, and normalise the skeleton data to facilitate feature extraction by TSFF-GCN. If the keypoint is not present in the 5 frames before and after the frame where the keypoint is missing, no compensation is performed.

The compensated skeleton data is sent to TSFF-GCN for feature extraction. Since the skeleton data has been normalised during the keypoint compensation stage,

TSFF-GCN does not need to perform batch normalisation on the skeleton data. Finally, action recognition is performed based on the extracted features using a Softmax classifier, and the recognition results are visualised using OpenCV.

$$p_i = \text{softmax}(f^i) \quad (17)$$

where p_i denotes the probability of action prediction, while f^i represents the extracted action features.

4 Experimental results and analyses

All experiments in this study were trained on the Pytorch 1.0 platform using three GTX 1080 Ti GPUs. During training, the algorithm was optimised using stochastic gradient descent with a momentum of 0.9 and a weight decay of 0.04.

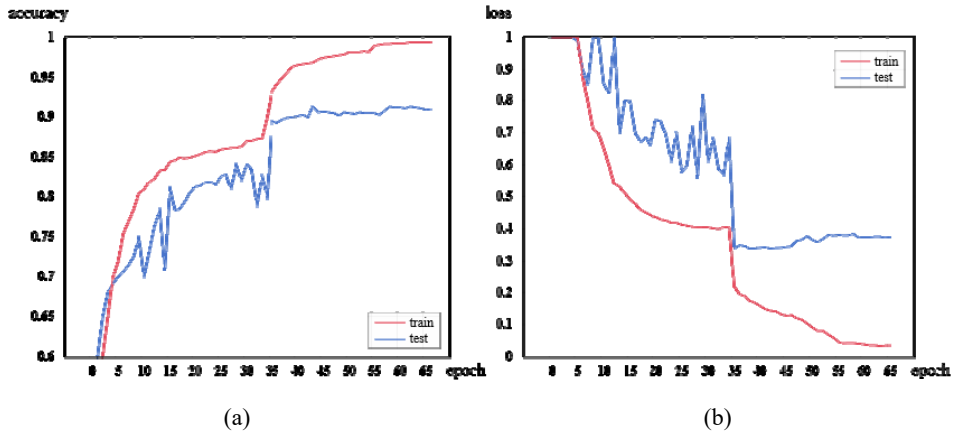
4.1 Effectiveness analysis of TSFF-GCN

This study employs multiple ablation experiments to validate the performance and efficiency of the proposed TSFF-GCN. The model is evaluated on the public dataset NTU-RGB+D 120 (Liu et al., 2020) and compared with state-of-the-art methods to demonstrate its effectiveness. The experimental setup is as follows: batch size is set to 128, training cycles to 65, and the initial learning rate to 0.1. At cycles 35 and 55, the learning rate is divided by 10. A warm-up strategy is employed for the first five cycles to stabilise the training process. Data preprocessing adopts the SGN approach; if a single frame of action samples contains two individuals, it is decomposed into two frames, each containing the action of one person. During training, the entire skeleton sequence was divided into 20 equal sub-sequences. One frame was randomly selected from each sub-sequence to generate a new sequence comprising 20 frames. Additionally, random rotations were applied to skeleton sequences for data augmentation, enhancing robustness to viewpoint variations. For testing, five new sequences were randomly generated, and the average score method was used to predict action categories.

Figure 5(a) illustrates the relationship between training epochs and recognition accuracy, with the horizontal axis representing training epochs and the vertical axis denoting recognition accuracy. The orange curve depicts the training process, while the blue curve represents the testing process. At epoch 35, the model's learning rate decreased from 0.1 to 0.01, leading to a rapid improvement in accuracy. Although the training accuracy subsequently exceeded 95% and gradually increased, approaching 100% after 50 epochs, the testing accuracy consistently remained between 90% and 90.8%, representing the model's upper limit. Figure 5(b) exhibits the same trend, with the horizontal axis denoting the number of cycles and the vertical axis representing model loss. At the 35th cycle, model loss drops sharply. The training process loss falls below 0.1 after 45 cycles, while the testing process loss remains consistently around 0.36 after the 35th cycle.

Regarding the experimental content, first, to assess the impact of individual modules, we performed a series of ablation studies. These were performed on the NTU RGB+D 60 dataset using joint data under the X-Sub (Cross-Subject) benchmark.

Figure 5 Performance and convergence speed of TSFF-GCN, (a) accuracy (b) loss value (see online version for colours)



For evaluating the effectiveness of spatial temporal fusion strategies, this subsection selected CTR-GCN as the baseline model and designed two additional models for comparison. Specifically, STA-GCN employs three spatial temporal aggregation convolutions in parallel solely within the spatial dimension to extract dynamic spatial temporal features. FF-GCN V1 replaces the three spatial temporal aggregation convolutions in STA-GCN with feature fusion convolutions, merging the output of each spatial temporal aggregation convolution with prior topological features. FF-GCN V2 modifies FF-GCN V1 by altering the feature fusion position: it first sums the outputs of the three spatial temporal aggregation graph convolutions from STA-GCN before performing feature fusion with the prior topological features. STF-GCN adopts the spatial temporal feature fusion module proposed in this chapter, incorporating TCN in addition to FF-GCN V2. Performance evaluations were conducted on these models, with experimental results presented in Table 1. Compared to the baseline model CTR-GCN, STA-GCN more effectively utilises spatial temporal features by aggregating spatial temporal information, thereby improving recognition accuracy. Comparisons of recognition accuracy between STA-GCN and both FF-GCN V1 and FF-GCN V2, as well as between FF-GCN V1 and FF-GCN V2, reveal that incorporating prior topological feature fusion can enhance model performance to a certain extent.

Table 1 Action recognition accuracy rates across different models

Model	X -Sub(%)
CTR-GCN	89.76
STA-GCN	89.97(+0.21)
FF-GCN V1	90.02(+0.26)
FF-GCN V2	90.23(+0.47)
STF-GCN	90.44(+0.68)

To evaluate the effectiveness of the extended temporal strategy, this experiment first selected STF-GCN as the baseline model. Building upon this, two distinct models were designed to validate the superiority of the structural design proposed in this chapter.

Among these, STF-GCN(TCN) replaces the original MSTC module with TCN and employs a 9×1 convolutional kernel. STF-GCN(+SE) incorporates an SE module onto STF-GCN, while STF-GCN(+eSE) represents the proposed model (TSFF-GCN). STF-GCN(+FCAM) extends STF-GCN by adding a Frequency-driven Channel Attention Module (FCAM) (Li et al., 2022), which comprises three operations: compression, excitation, and feature recalibration. For compression, FCAM employs 1DGAP and 1D DCT to simultaneously compress spatial and temporal dimensions, while excitation and feature recalibration operations align with the eSE module. This section conducts performance evaluations on these five models, with results presented in Table 2. It can be observed that compared to the baseline model, STF-GCN(TCN) exhibits a performance decline. This is attributable to TCN’s limitation in employing only fixed convolution kernels, which restricts its ability to flexibly model dynamic temporal sequences. Among the three models applying attention mechanisms, the model proposed in this chapter demonstrates the highest recognition accuracy. The SE module employs two fully connected layers during excitation to first reduce and then increase dimensionality. Whilst this approach reduces model parameters, it may cause feature maps to lose critical channel information due to diminished channel dimensions, thereby compromising model performance. FCAM requires multiple applications of frequency components to compress temporal information, slowing model convergence. The experimental results in Table 2 further validate the effectiveness of the extended temporal strategy proposed in this section.

Table 2 Action recognition accuracy of different models

<i>Method</i>	<i>X-Sub (%)</i>	<i>Parametric quantity (M)</i>	<i>FLOPS (G)</i>
STF-GCN	90.44	2.74	3.23
STF-GCN(TCN)	90.27	4.67	5.42
STF-GCN(+SE)	90.43	2.87	3.42
STF-GCN(+eSE)	90.59	3.00	3.57
STF-GCN(+FCAM)	90.17	2.87	3.76

Secondly, to validate the effectiveness of the TSFF-GCN model proposed in this study, we conducted evaluations on the NTURGB+D 120 dataset under the X-Sub and X-Set (Cross-Set) benchmarks using joint data. Comparisons were made with state-of-the-art behaviour recognition methods, with results presented in Table 3.

Table 3 Comparison of accuracy between TSFF-GCN and other methods on the NTU RGB+D 120 dataset

<i>Method</i>	<i>X-Sub (%)</i>	<i>X-Set (%)</i>
TSTN (Wang et al., 2023)	85.1	86.8
LKA-GCN (Liu et al., 2023)	86.3	87.8
MSSTNet (Cheng et al., 2023)	87.4	88.3
EfficientGCN-B4 (Song et al., 2023)	88.7	89.1
JD-TC (Gao et al., 2023)	88.8	90.1
TSFF-GCN	89.1	90.4

Through performance comparisons, it is evident that the proposed TSFF-GCN demonstrates outstanding capabilities. Compared to other models in the table, TSFF-GCN's spatial temporal feature fusion strategy enables the model to model dynamic skeleton sequences, fuse spatial temporal features, and generate dynamic spatial temporal topologies. This endows TSFF-GCN with exceptional generalisation performance and expressive power. Moreover, TSFF-GCN's extended temporal strategy enables comprehensive extraction of sequential dynamics, further enhancing its performance. Collectively, these characteristics confer significant advantages to TSFF-GCN across action recognition tasks on diverse datasets.

4.2 Analysis of the effectiveness of recognising dance movements in student instruction

Furthermore, to evaluate the method's recognition performance in student dance instruction, this chapter constructed the DanceTeach dataset specifically designed for this scenario. It comprises 200 dance movement videos across 10 categories, with durations ranging from 6.3 to 15.2 seconds. The videos were captured in well-lit indoor environments free from distractions. The dataset includes ten dance movements: *daoti zijinguan* (backward kick to the crown), *yunli* (moving through clouds), *quzhou qianqiao* (elbow-bent forward bridge), *yanshi zijinguan* (swallow-style kick to the crown), *woyu* (lying fish), *chuaiyan* (kicking swallow), *xuanzi* (spinning jump), *limao fanshen* (wildcat turn), *yuanbao* (ingot pose), and *datiao* (grand leap). Each dance movement comprises five video clips from different dancers, each recorded at 25 frames per second, annotated according to the specific dance movement performed in each clip. The training set comprises 140 action videos, while the test set contains 60 action videos (comprising 5 types, 12 videos per type), maintaining a training-to-test ratio of 7:3. The model's initial learning rate is set to 0.05, with a random dropout rate of 50%. The recognition accuracy of TSFF-GCN for the ten dance movements in the DanceTeach dataset is presented in Table 4, demonstrating the model's effectiveness for this task.

Table 4 Performance of TSFF-GCN on the DanceTeach dataset

<i>Dance movements</i>	<i>TSFF-GCN (%)</i>
daoti zijinguan	92.9
yunli	96.1
quzhou qianqiao	93.7
yanshi zijinguan	94.3
woyu	91.6
chuaiyan	89.6
xuanzi	94.9
limao fanshen	93.4
yuanbao	91.7
datiao	94.2

5 Conclusions

This study proposes an extended temporal and spatial temporal feature fusion graph convolutional network specifically tailored for student dance instruction action recognition and classification. The proposed method successfully addresses limitations of existing approaches through innovative architectural design, incorporating spatial temporal feature fusion modules and extended temporal convolutional components. The spatial temporal feature fusion module effectively integrates dynamic features through parallel graph convolutions and attention-based mechanisms, whilst the extended temporal module employs dilated convolutions to capture multi-scale temporal patterns. Experimental validation on a dance instruction dataset comprising 200 video sequences across 10 classical dance categories demonstrates superior recognition performance compared to conventional approaches. This research provides a specialised framework for dance education applications and lays the groundwork for developing intelligent dance instruction systems. Future research directions include expanding the dataset to encompass broader dance styles and investigating real-time implementation for interactive teaching environments.

Declarations

All authors declare that they have no conflicts of interest.

References

- Al-Faris, M., Chiverton, J., Ndzi, D. and Ahmed, A.I. (2020) ‘A review on computer vision-based methods for human action recognition’, *J. Imaging*, Vol. 6, No. 6, p.46.
- Alam, A. and Mohanty, A. (2023) ‘Educational technology: exploring the convergence of technology and pedagogy through mobility, interactivity, AI, and learning tools’, *Cogent Engineering*, Vol. 10, No. 2, p.2283282.
- Bian, C., Feng, W., Wan, L. and Wang, S. (2021) ‘Structural knowledge distillation for efficient skeleton-based action recognition’, *IEEE Trans. Image Process.*, Vol. 30, pp.2963–2976.
- Cheng, Q., Cheng, J., Ren, Z., Zhang, Q. and Liu, J. (2023) ‘Multi-scale spatial-temporal convolutional neural network for skeleton-based action recognition’, *Pattern Analysis and Applications*, Vol. 26, No. 3, pp.1303–1315.
- Dimitriadou, E. and Lanitis, A. (2024) ‘Student action recognition for improving teacher feedback during tele-education’, *IEEE Transactions on Learning Technologies*, Vol. 17, pp.569–584.
- Ding, X., Peng, W. and Yi, X. (2022) ‘Evaluation of physical education teaching effect based on action skill recognition’, *Comput. Intell. Neurosci.*, Vol. 2022, p.9489704.
- Gao, X., Du, S. and Yang, Y. (2023) ‘Glimpse and focus: global and local-scale graph convolution network for skeleton-based action recognition’, *Neural Netw.*, Vol. 167, pp.551–558.
- George, B. and Wooden, O. (2023) ‘Managing the strategic transformation of higher education through artificial intelligence’, *Administrative Sciences*, Vol. 13, No. 9, pp.196–215.
- Guo, H., Zou, S., Xu, Y., Yang, H., Wang, J., Zhang, H. and Chen, W. (2021) ‘DanceVis: toward better understanding of online cheer and dance training’, *Journal of Visualization*, Vol. 25, No. 1, pp.159–174.
- Kuchynska, I., Orlov, O., Kobysia, A., Kutsak, L. and Yesselbayeva, A. (2025) ‘Adaptation of pedagogical approaches to the implementation of intelligent systems in the educational process’, *Periodicals of Engineering and Natural Sciences (PEN)*, Vol. 13, No. 1, pp.281–294.

- Lai, Z., Yang, J., Xia, S., Wu, Q., Sun, Z., Yu, W. and Pei, L. (2025) 'SMART: scene-motion-aware human action recognition framework for mental disorder group', *IEEE Internet of Things Journal*, Vol. 12, No. 8, pp.10099–10113.
- Li, F., Zhu, A., Li, J., Xu, Y., Zhang, Y., Yin, H. and Hua, G. (2022) 'Frequency-driven channel attention-augmented full-scale temporal modeling network for skeleton-based action recognition', *Knowledge-Based Systems*, Vol. 256, p.109854.
- Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L.Y. and Kot, A.C. (2020) 'NTU RGB+D 120: a large-scale benchmark for 3D human activity understanding', *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 42, No. 10, pp.2684–2701.
- Liu, Y., Zhang, H., Li, Y., He, K. and Xu, D. (2023) 'Skeleton-based human action recognition via large-kernel attention graph convolutional network', *IEEE Trans. Vis. Comput. Graph.*, Vol. 29, No. 5, pp.2575–2585.
- Mao, L. (2021) 'Remote classroom action recognition based on improved neural network and face recognition', *Journal of Intelligent & Fuzzy Systems*, Vol. 20, pp.1–11.
- Pareek, P. and Thakkar, A. (2020) 'A survey on video-based human action recognition: recent updates, datasets, challenges, and applications', *Artificial Intelligence Review*, Vol. 54, No. 3, pp.2259–2322.
- Pastukhov, O. (2021) 'Modern dance: theoretical foundations of the latest approaches to teaching', *National Academy of Managerial Staff of Culture and Arts Herald*, Vol. 31, No. 4, pp.106–110.
- Qin, W. and Meng, J. (2025) 'The research on dance motion quality evaluation based on spatiotemporal convolutional neural networks', *Alexandria Engineering Journal*, Vol. 114, pp.46–54.
- Song, Y.F., Zhang, Z., Shan, C. and Wang, L. (2023) 'Constructing stronger and faster baselines for skeleton-based action recognition', *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 45, No. 2, pp.1474–1488.
- Wang, H., Kläser, A., Schmid, C. and Liu, C.-L. (2013) 'Dense trajectories and motion boundary descriptors for action recognition', *International Journal of Computer Vision*, Vol. 103, No. 1, pp.60–79.
- Wang, L., Zhang, J., Yang, S. and Gu, S. (2023) 'Two-stream spatiotemporal networks for skeleton action recognition', *IET Image Processing*, Vol. 17, No. 11, pp.3358–3370.
- Wang, Q., Zhang, K. and Asghar, M.A. (2022) 'Skeleton-based ST-GCN for human action recognition with extended skeleton graph and partitioning strategy', *IEEE Access*, Vol. 10, pp.41403–41410.
- Wang, X., Zhang, W., Wang, C., Gao, Y. and Liu, M. (2024) 'Dynamic dense graph convolutional network for skeleton-based human motion prediction', *IEEE Trans. Image Process.*, Vol. 33, pp.1–15.
- Wang, Y. and Jan, N. (2022) 'Research on dance movement recognition based on multi-source information', *Mathematical Problems in Engineering*, Vol. 2022, pp.1–10.
- Wu, F., Wang, Q., Bian, J., Ding, N., Lu, F., Cheng, J., Dou, D. and Xiong, H. (2023) 'A survey on video action recognition in sports: datasets, methods and applications', *IEEE Trans. Multimedia*, Vol. 25, pp.7943–7966.
- Yang, H., Yan, D., Zhang, L., Sun, Y., Li, D. and Maybank, S.J. (2022) 'Feedback graph convolutional network for skeleton-based action recognition', *IEEE Trans. Image Process.*, Vol. 31, pp.164–175.
- Zhang, D., Wang, H., Weng, C. and Shi, X. (2021) 'Video human action recognition with channel attention on ST-GCN', *Journal of Physics: Conference Series*, Vol. 2010, No. 1, p.012131.
- Zhang, L. and Li, Q. (2022) 'Improved graph convolutional neural network for dance tracking and pose estimation', *Computational Intelligence and Neuroscience*, Vol. 2022, pp.1–10.