Detection of Climbing Rod Motion Changes Based on 3D Pose Estimation

Heping Zheng[1,*], Yudong Gan[1], Chenqingyu Zhang[1], Shuang Liang[1], Shengxia Zhao[1], Huangjing Zhang[1]

[1]State Grid Sichuan Electric Power Company Skills Training Center, Chengdu 611133, Sichuan, China

Corresponding author: Heping Zheng (e-mail: renbingcai3@163.com).

ABSTRACT 3D human pose estimation has important engineering application value in industrial scenarios such as safety monitoring of power grid pole-climbing operations and standardized action training. Existing skeleton-based human behavior recognition methods still have notable limitations in modeling long-term temporal dependencies, capturing multi-scale dynamic motion patterns, and resolving geometric ambiguities in 2D-to-3D pose lifting. This paper proposes a graph-based network integrating multi-scale temporal convolution with spatial attention, and an interactive spatio-temporal Transformer, and develops an end-to-end web-based recognition system. Experiments on public benchmark datasets fully verify the performance of the proposed methods, which can provide technical support for skeleton-based action recognition and deployment of intelligent perception systems in industrial scenarios.

INDEX TERMS Human action recognition; 3D human pose estimation; graph convolutional network; Transformer; multi-scale modeling; attention mechanism

I. INTRODUCTION

Pole climbing is a high-risk operation in power-grid maintenance, communication-tower inspection, and professional sports training. Detecting changes in such movements imposes stringent requirements on the accuracy and robustness of 3D human pose estimation, while poor illumination, cluttered backgrounds, and frequent self-occlusion in complex outdoor environments further increase the technical difficulty.

Existing skeleton-based action-recognition methods generally adopt a two-stage pipeline of "2D keypoint detection $\rightarrow$ 3D pose lifting $\rightarrow$ spatio-temporal feature learning $\rightarrow$ action classification," but they are affected by inherent depth ambiguity caused by perspective projection. Mainstream spatio-temporal graph convolutional networks (ST-GCNs) are constrained by fixed temporal convolution kernels. In power-operation action-recognition scenarios, they therefore struggle to capture both short-term rapid limb movements and long-term center-of-gravity adjustments during pole climbing. They also lack an effective mechanism for focusing on salient features and make insufficient use of local joint context [1]. Transformer-based methods are effective at modeling long-range temporal dependencies, but they commonly process the spatial and temporal dimensions separately and fail to fully exploit structured constraints among human joints and inter-frame temporal consistency. This both weakens their ability to resolve depth ambiguity and increases computational cost.

In terms of motion characteristics and technical challenges, pole-climbing operations share substantial commonality with the actions covered by the Human3.6M benchmark dataset. Pole climbing includes periodic continuous movements with alternating limbs, which are consistent with the temporal motion patterns of actions such as walking and walking a dog in the dataset. Dynamic adjustments of the trunk's center of gravity and changes in limb posture correspond to the spatial joint-variation patterns of actions such as sitting down and posing. Local actions such as upper-limb gripping and exertion are also consistent with the characteristics of upper-limb-dominant actions such as taking photos and smoking. Both types of scenario face common technical challenges, including joint self-occlusion, depth ambiguity, and the coexistence of multi-scale motions; these are precisely the core issues addressed by the proposed method. In 3D human pose estimation, it is common practice in both academia and industry to first validate an algorithm's spatio-temporal modeling capability and accuracy on public benchmark datasets and then perform transfer optimization for a specific industrial scenario. Therefore, validation on Human3.6M is scientifically sound and comparable and provides a reliable performance benchmark for subsequent deployment in pole-climbing scenarios.

To address these issues, this paper proposes an end-to-end integrated 3D skeleton-based action-recognition framework for detecting changes in pole-climbing movements. The framework uses a pretrained OpenPose model to extract 2D pose sequences; designs a Spatial Multi-scale Temporal Graph Convolutional Attention Network (SMT-GCA) to

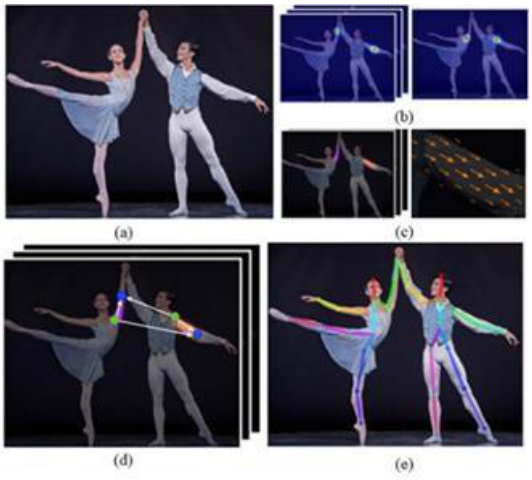

FIGURE 1. Overview of the OpenPose pipeline: (a) Input image; (b) Part confidence maps; (c) Part Affinity Fields (PAFs); (d) Bipartite matching; (e) Output results.

learn discriminative spatio-temporal features; develops an Interactive Spatio-Temporal Transformer (ISTFormer) for high-accuracy 2D-to-3D pose lifting; and constructs an end-to-end web-based system that supports the complete workflow from monocular video input to action classification and movement-change detection.The proposed method significantly outperforms the baseline models on public benchmark datasets and provides a practical technical solution for industrial safety monitoring.

II. TECHNICAL BACKGROUND

A. 2D POSE ESTIMATOR

This study focuses on skeleton-based human action recognition and 2D-to-3D pose lifting. Therefore, the high-accuracy and fast 2D pose estimator OpenPose is used to extract keypoint locations and the corresponding 2D skeleton structures from video sequences [2].

OpenPose is a real-time multi-person 2D pose-estimation system capable of simultaneously detecting keypoints on the human torso, feet, hands, and face in a single image. As shown in Fig. 1, a color image of size $w \times h$ is fed into a feed-forward neural network that predicts two outputs: 2D confidence maps $S = (S_1, S_2, \ldots, S_J)$, representing the likelihood of each keypoint location, and Part Affinity Fields $L = (L_1, L_2, \ldots, L_C)$, which encode directional associations between connected body parts. Among them, J represents the number of key points, $S_j \in \mathbb{R}^{w \times h}$ ($j \in \{1, \ldots, J\}$); C represents the number of limb connections, $L_c \in \mathbb{R}^{w \times h \times 2}$ ($c \in \{1, \ldots, C\}$). Subsequently, the bipartite graph matching algorithm based on the Hungarian algorithm associates keypoints into individual skeletons by jointly analyzing confidence maps and component affinity fields. The final output is a two-dimensional pose sequence for each detected person in the scene.

The core idea of the Hungarian algorithm is to iteratively search for augmented paths, gradually expanding the matching scale until no more augmented paths can be found. At each step, the algorithm attempts to increase the number of matches through alternating paths and continues until no further augmentation is possible, thereby obtaining a maximum matching [3].

Based on the Hungarian algorithm, the keypoints corresponding to each individual are correlated to form the final two-dimensional pose.

B. GRAPH CONVOLUTIONAL NETWORK

CNN is only applicable to regular two-dimensional structured data such as images, and cannot directly process non-Euclidean graph structured data such as human skeletons.

Graph Convolutional Network (GCN) is used to process non-Euclidean data by defining convolution operations on the graph to extract features, which can effectively handle human skeleton topology data [4].

For an undirected graph with nodes $G = (V, E)$, use an adjacency matrix $A \in \mathbb{R}^{N \times N}$ to encode the connection relationships between nodes: $A_{ij} = 1$ represents nodes v_i and v_j and there are edges between them, otherwise $A_{ij} = 0$. The eigenvector $X \in \mathbb{R}^{N \times D}$ represents the D-dimensional eigenvectors of each node.

Graph convolution aggregates the features of nodes and their neighbors. The graph convolution formula for the $(l+1)$ layer is shown in equation (1):

$$H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)}\right) \tag{1}$$

Among them, $H^{(l+1)} \in \mathbb{R}^{N \times D}$ represents the feature matrix of the l-th layer, $H^{(0)} = X$, $W^{(l)}$ represents learnable weights, and $\sigma(\cdot)$ is a nonlinear activation function.

Because the diagonal elements of adjacency matrix A are zero, the node's own features would be ignored. In addition, differences in node degree may shift the distribution of aggregated features. Therefore, an identity matrix I_N is added to A to obtain the augmented adjacency matrix $\widetilde{A} = A + I_N$, followed by symmetric normalization. The final graph-convolution formula is given in Eq. (2):

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

Among them, $\widetilde{D}$ is the degree matrix of $\widetilde{A}$, defined as $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$; $\widetilde{D}^{-1/2}\widetilde{A}\widetilde{D}^{-1/2}$ is the normalized adjacency matrix used in Eq. (2).

C. SPATIOTEMPORAL GRAPH CONVOLUTION MODEL

The Spatio Temporal Graph Convolutional Network (ST-GCN) models human motion by constructing a spatiotemporal skeleton sequence graph. The node set $V = \{v_{ti} \mid t = 1, \ldots, T;\ i = 1, \ldots, N\}$ contains all the joints in the skeleton sequence, where T represents the number of frames in the input video, N represents the number of joints in the single person skeleton image, and v_{ti} represents the i-th joint in the t-th frame. The edge set E consists of two subsets: $E_S = \{v_{ti}v_{tj} \mid (i, j) \in H\}$ containing

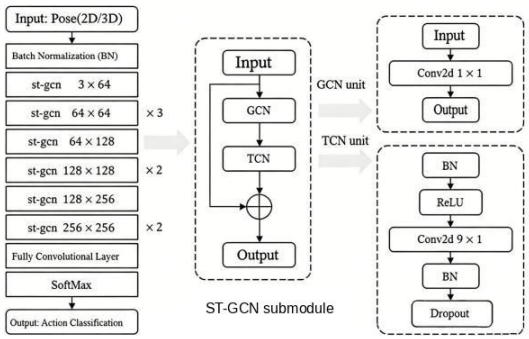

FIGURE 2. Network architecture of the spatio-temporal graph convolutional model.

connections between joints within the same frame, where H is the set of joints naturally connected by the human body; $E_F = \{v_{ti}v_{(t+1)i}\}$ captures the connections of the same joint between adjacent frames. Therefore, all edges of a specific joint i in E_F represent the motion trajectory of that node over time.

The overall process of behavior recognition is as follows: first, construct a spatiotemporal skeleton map, extract features through multiple layers of ST-GCN, and finally output action category probability vectors through a SoftMax classifier [6].

The detailed architecture of the model is shown in Figure 2: After batch normalization of the input, the channel is first projected to 64 dimensions through one ST-GCN submodule, and then features are extracted layer by layer through nine submodules with residual connections. Finally, the classification results are output through global average pooling and fully connected layers.

Nine ST-GCN submodules are divided into 3 groups according to output channels (64/128/256 dimensions), the 4th and 7th layers adopt a downsampling with a step size of 2 Reduce the time resolution. Each submodule is composed of spatial GCN units A Temporal Convolutional Network with a Kernel Size of 9×1 Convolutional Network (TCN) unit composition [7], with 0.5 Probability Dropout prevents overfitting by using residual connections to integrate inputs Input information, layer representation as shown in equation (3):

$$H^{(l+1)} = f_{\text{stgcn}}\left(H^{(l)}\right) = f_{\text{TCN}}\left(f_{\text{GCN}}\left(H^{(l)}\right)\right) + H^{(l)} \tag{3}$$

Among them, $H^{(l)}$ represents the feature output of the l-th layer ST-GCN submodule.

D. VISUAL TRANSFORMER

The architecture of Vision Transformer (ViT) is shown in Figure 3. ViT divides the input image into fixed patches, and combines linear embedding and positional embedding to form a feature sequence. The input Transformer encoder models the patch relationship and outputs the classification result [8].

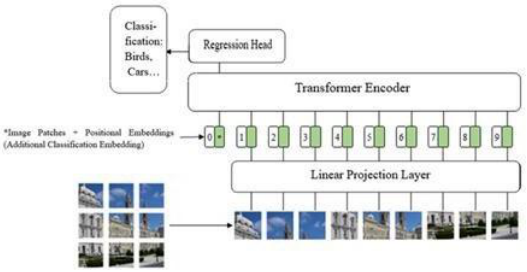

FIGURE 3. Vision Transformer Model.

The Transformer encoder architecture consists of alternating Multi Head Self Attention (MSA) and Multi Layer Perceptron (MLP) modules, with Layer Normalization (LN) applied before each module and residual connections used afterwards.

MSA adopts scaled dot product attention, integrates the features of query matrix Q, key matrix K, and value matrix V, and outputs an attention matrix. Among them, $Q, K, V \in \mathbb{R}^{N\times d}$, N is the number of vectors in the sequence, and d is the dimension of each vector. In this attention operation, the scaling factor $1/\sqrt{d}$ prevents excessively large dot products and vanishing gradients during training. Therefore, the output of scaled dot product attention is shown in equation (4):

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{4}$$

The matrices Q, K, and V are calculated from the input feature Z through learnable linear transformations W_Q, W_K, and W_V, as shown in equation (5):

$$Q = ZW_Q, \quad K = ZW_K, \quad V = ZW_V \tag{5}$$

MSA models information in different subspaces through multiple attention heads, which independently calculate attention and concatenate projections. The output is shown in equation (6):

$$\text{MSA}(Q, K, V) = \text{Concat}(H_1, H_2, \ldots, H_h)W_{\text{out}} \tag{6}$$

Among them, $H_i = \text{Attention}(Q_i, K_i, V_i)$, $i \in [1, \ldots, h]$, where h represents the number of attention heads.

The input of the Multi Layer Perceptron (MLP) module is linearly projected to high dimensions and activated through GELU Dropout, The second linear layer and Dropout output.

The formula for the l-th layer of the Transformer encoder module is as follows:

$$Z'_l = \text{MSA}(\text{LN}(Z_{l-1})) + Z_{l-1}, \quad l = 1, 2, \ldots, L \tag{7}$$

$$Z_l = \text{MLP}(\text{LN}(Z'_l)) + Z'_l, \quad l = 1, 2, \ldots, L \tag{8}$$

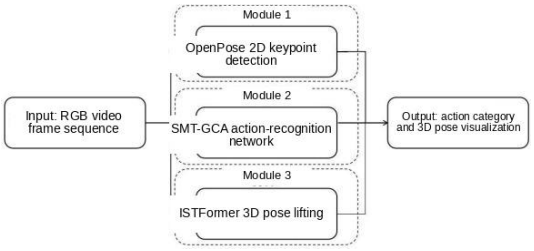

FIGURE 4. Overall architecture of the proposed end-to-end 3D pose estimation framework.

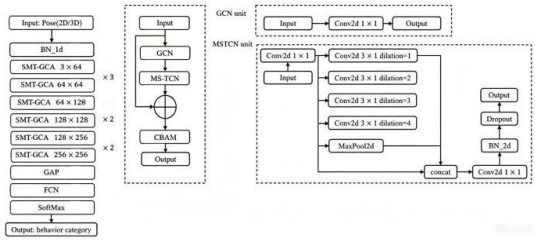

FIGURE 5. Overall flowchart of the SMT-GCA algorithm.

$$Y = \mathrm{LN}(Z_L) \tag{9}$$

Among them, Z_l represents the fused feature sequence input to the l-th encoder layer; $\mathrm{LN}(\cdot)$ represents layer normalization; L is the total number of Transformer encoder layers; $Y = Z_L$ is the output feature sequence after L encoder layers.

III. END TO END 3D POSE ESTIMATION METHOD COMBINING SMT-GCA AND ISTFORMER

The end-to-end framework proposed in this article consists of four core components: pretrained OpenPose 2D keypoint detector, SMT-GCA network, ISTFormer module, and end-to-end web-based behavior recognition system. OpenPose first generates a normalized two-dimensional joint coordinate sequence and constructs a spatiotemporal skeleton map from the RGB video frame sequence collected for outdoor industrial operation scenarios; SMT-GCA network extracts hierarchical multi-scale motion features and highlights key joint information; ISTFormer elevates 2D pose sequences to precise 3D representations by modeling cross joint spatial interactions and cross frame temporal consistency; Finally, the web-based system integration results in real-time action classification and detection of abnormal motion changes. The overall data flow and module connection relationship of the end-to-end framework are shown in Figure 4.

A. CONSTRUCTION OF SMT-GCA NETWORK FOR SKELETON ACTION RECOGNITION

ST-GCN models joint topology associations using spatial GCN, extracts inter frame features using single size temporal convolutions, and forms a deep network by stacking residual connections. It is the mainstream baseline model in the field of skeleton action recognition. But it has two core inherent flaws: firstly, the temporal modeling only uses fixed size single scale convolution kernels, which makes it difficult to simultaneously capture multi-scale patterns such as short-term subtle limb movements and long-term posture adjustments in industrial operation actions; The second is to uniformly aggregate all joint features, lacking an attention focusing mechanism for key discriminative joints, which is susceptible to interference from redundant joint information and occlusion noise [9].

In response to the above shortcomings, this paper proposes the Spatial Multiscale Temporal Graph Convolutional Attention Network (SMT-GCA), which retains the modeling ability of ST-GCN spatial graph convolution skeleton and residual connection structure, and makes two core improvements: first, upgrading the temporal module: replacing the original single scale TCN with a Multi Scale Temporal Convolutional Network (MSTCN), capturing motion features of different temporal granularities in parallel through multi branch dilated convolution, balancing short-term action details and long-term motion dependencies; The second is the introduction of attention mechanism: a new Convolutional Block Attention Module (CBAM) is added, which adaptively weights joint features from both channel and spatial dimensions, enhances attention to discriminative joints of actions, and suppresses irrelevant information interference.

The core submodule of SMT-GCA consists of spatial graph convolution, MSTCN, and CBAM connected in series. The overall network flow is shown in Figure 5. Compared to the classic ST-GCN architecture in Figure 2 SMT-GCA replaces single temporal convolution units and adds attention modules in each residual submodule, enhancing feature representation while maintaining network depth and topological modeling capabilities. After batch normalization of the input, the channel is first increased from 3 dimensions to 64 dimensions through one SMT-GCA submodule without residual connections. Then, spatiotemporal features are learned through nine submodules with residual connections. Finally, the classification results are output through global average pooling and a fully convolutional network [10].

Construct a skeleton spatiotemporal sequence diagram $G = (V, E)$ based on the two-dimensional joint coordinates output by OpenPose. Among them, $V = \{v_{ti} \mid t = 1, \ldots, T,\ i = 1, \ldots, N\}$ is the set of all joints in the sequence diagram, T is the number of input video frames, and N is the number of joints in a single frame. $E = \{E_S, E_F\}$ is the set of all connected edges in the sequence diagram, where E_S represents the set of connected edges of joints within a frame, and E_F represents the set of connected edges of the same joint between frames [11]. This allows us to represent the motion trajectory of each joint in the skeleton over time.

The graph convolution requires redefining the sampling function and weight function. In the figure, the sampling function is defined on the neighborhood set $B(v_{ti}) = \{v_{tj} \mid$

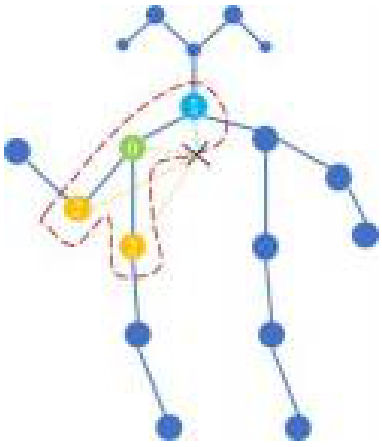

FIGURE 6. Spatial configuration partitioning strategy.

$d(v_{tj}, v_{ti}) \leq D\}$ of node v_{ti}, where $d(v_{tj}, v_{ti})$ represents the minimum value of any path from node v_{tj} to node v_{tj}. If $D = 1$, then the neighborhood $B(v_{ti})$ of the node is $d(v_{tj}, v_{ti}) = v_{tj}$. At this point, the redefinition of the sampling function p is shown in equation (10) [12]:

$$p(v_{ti}, v_{tj}) = v_{tj} \tag{10}$$

This study adopts the spatial configuration partitioning strategy shown in Fig. 6. The neighborhood of each node is divided into three subsets: the root node itself, a centripetal set (closer to the skeleton's center of gravity), and a centrifugal set (farther from the center of gravity), with each subset assigned a different convolution weight.

The partitioning rule is shown in Equation (11).

$$l_{ti}(v_{tj}) = \begin{cases} 0, & r_i = r_j, \\ 1, & r_j < r_i, \\ 2, & r_j > r_i. \end{cases} \tag{11}$$

Where r_i is the average distance from the center of gravity to joint i, and r_j is the distance from joint j to the center of gravity.

Using this partitioning strategy, nodes in the neighborhood are mapped to subset labels $l_{ti} : B(v_{ti}) \rightarrow \{0, \ldots, K-1\}$. The weighting function w is redefined as shown in Eq. (12):

$$w(v_{ti}, v_{tj}) = w'(l_{ti}(v_{tj})) \tag{12}$$

After redefining the sampling and weighting functions, the spatial GCN is expressed by Eq. (13):

$$f_{\text{out}}(x) = \sum_{v_{tj} \in B(v_{ti})} \frac{1}{Z_{ti}(v_{tj})} f_{\text{in}}(v_{tj}) \cdot w'(l_{ti}(v_{tj})) \tag{13}$$

The spatial GCN is extended to the spatio-temporal domain, where the neighborhood of a node is defined in Eq. (14):

$$B(v_{ti}) = \left\{ v_{qj} \mid d(v_{tj}, v_{ti}) \leq K,\ |q-t| \leq \left\lfloor \frac{\Gamma}{2} \right\rfloor \right\} \tag{14}$$

where K is the spatial-distance threshold and Γ is the temporal convolution-kernel size.

The node-label mapping in the spatio-temporal domain is given in Eq. (15):

$$l_{\text{ST}}(v_{qj}) = l_{ti}(v_{tj}) + \left(q - t + \left\lfloor \frac{\Gamma}{2} \right\rfloor \right) K \tag{15}$$

where $l_{ti}(v_{ti})$ represents the label mapping result for internal node v_{ti} within a single frame. This completes the construction of the spatio-temporal graph convolution.

As the first major improvement of SMT-GCA over ST-GCN, MSTCN addresses the modeling limitations of single-scale temporal convolution. The TCN subunit in a conventional ST-GCN uses only a one-dimensional temporal convolution with a kernel size of 9×1 and a stride of 1, which leads to two main problems [13]. First, the large kernel is not effective at capturing subtle short-term motion changes. Second, a fixed-size convolution kernel cannot adapt to motion patterns at different temporal scales. Existing studies show that multi-branch dilated temporal convolution can effectively capture action patterns of different durations and is a mainstream solution to the limitations of a single temporal scale [14].

Dilated convolution is introduced to improve the temporal-convolution module. By changing the dilation rate, receptive fields of different scales can be obtained without increasing the number of parameters, while the output feature-map size remains unchanged. The number of branches and the dilation-rate range are designed by jointly considering modeling capability, parameter count, and real-time performance.

If too few branches are used, the module cannot cover a sufficient range of temporal scales; too many branches cause parameter redundancy, greater computational cost, and a higher risk of overfitting. Referring to the common design paradigm of multi-scale temporal modules in skeleton-based action recognition and considering the temporal complexity of industrial operation actions, six parallel branches are ultimately adopted. This configuration provides adequate multi-scale coverage while controlling parameter count and inference latency, thereby satisfying the real-time requirements of outdoor edge deployment.

For the dilation-rate range, four dilated-convolution branches use dilation rates from 1 to 4. For a 3×1 convolution kernel, the corresponding effective receptive fields are 3, 5, 7, and 9 frames, covering temporal patterns from subtle single-frame limb adjustments to continuous multi-frame periodic movements. When the dilation rate exceeds 4, temporal sampling becomes excessively sparse, local action details are lost, and the effective receptive field may be insufficient for short input sequences, which

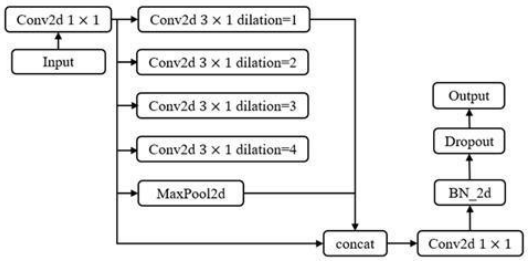

FIGURE 7. MSTCN unit.

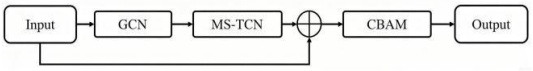

FIGURE 8. Spatial multi-scale temporal graph-convolution attention submodule after the introduction of CBAM.

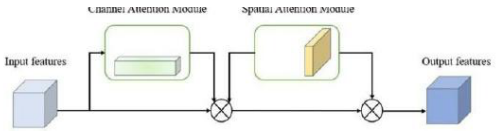

FIGURE 9. Flowchart of CBAM.

can reduce feature-extraction accuracy. An additional max-pooling branch strengthens salient motion features, while a 1×1 convolution branch preserves the original temporal information; together, they complement the dilated-convolution branches.

Based on these design principles, this study proposes MSTCN, which replaces the original single-scale temporal convolution with multi-branch temporal convolutions. The network architecture is shown in Fig. 7.

MSTCN first uses a 1×1 convolution to transform the channel dimension and divides the features into six equal-width channel groups: one max-pooling branch extracts salient temporal features; four 3×1 convolution branches use dilation rates of 1–4 to capture temporal dependencies at different scales; and a final 1×1 convolution fuses the outputs of all branches. Batch normalization and Dropout are then applied to obtain the final output.

As the second major improvement of SMT-GCA over ST-GCN, CBAM addresses the equal weighting of joint features and the insufficient prominence of discriminative information. Joint use of channel and spatial information is essential for improving skeleton-based human action recognition [15]. To enhance the extraction of features from key joints,

CBAM is introduced to form the spatial multi-scale temporal graph-convolution attention submodule shown in Fig. 8.

CBAM consists of two interacting submodules: a Channel Attention Module (CAM) and a Spatial Attention Module (SAM). The overall procedure is shown in Fig. 9.

The procedure of the channel-attention module is shown in Fig. 10. The input is an intermediate feature tensor $F \in \mathbb{R}^{C\times T\times V}$.

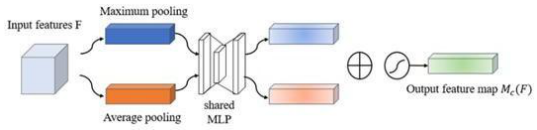

FIGURE 10. Channel-attention module flowchart.

Average pooling and max pooling compress the spatial dimensions, and the resulting features are processed by a shared MLP and added element by element. A Sigmoid activation then produces the channel-attention map. The calculation is given in Eq. (16):

$$\begin{aligned} M_C(F) &= \sigma(\mathrm{MLP}(\mathrm{AvgPool}(F)) + \mathrm{MLP}(\mathrm{MaxPool}(F))) \\ &= \sigma\big(W_1(\mathrm{ReLU}(W_0(F^c_{\mathrm{avg}}))) + W_1(\mathrm{ReLU}(W_0(F^c_{\mathrm{max}})))\big) \end{aligned} \tag{16}$$

Here, $\sigma(\cdot)$ is the sigmoid activation function; the descending weight in the MLP is $W_0 \in \mathbb{R}^{C/r\times C}$, and the ascending weight is $W_1 \in \mathbb{R}^{C\times C/r}$. These two weights are shared by the two inputs.

Perform element-wise multiplication between the output feature map $M_C(F)$ of the channel-attention module and the original CBAM input feature map F to generate the intermediate feature map $F' = F \odot M_C(F)$, which is then passed to the spatial-attention module. Max pooling and average pooling compress the channel dimension; the generated features are concatenated and passed through a 7×7 convolutional layer, after which a sigmoid activation function is applied to obtain the spatial-attention map $M_S(F')$. The calculation formula is shown in Equation (17):

$$\begin{aligned} M_S(F') &= \sigma\big(f^{7\times7}([\mathrm{MaxPool}(F'); \mathrm{AvgPool}(F')])\big) \\ &= \sigma\big(f^{7\times7}([F^s_{\mathrm{max}}; F^s_{\mathrm{avg}}])\big) \end{aligned} \tag{17}$$

Here, $f^{7\times7}$ denotes a two-dimensional convolution with a 7×7 kernel. The final CBAM output is calculated as (18) and (19):

$$F' = M_C(F) \odot F \tag{18}$$

$$F'' = M_S(F') \odot F' \tag{19}$$

Where $\odot$ denotes element-wise multiplication, F' is the final feature output.

B. CONSTRUCTION OF THE ISTFORMER MODEL FOR 3D POSE ESTIMATION

An architecture that integrates a Transformer with a graph convolutional network can simultaneously exploit the advantages of long-range temporal modeling and topology-aware representation and has become a mainstream approach to 2D-to-3D pose lifting [16]. The spatial Transformer module of the proposed ISTFormer extracts high-dimensional feature embeddings from a single-frame 2D skeleton pose. Its architecture is shown in Fig.11. The input is the 2D skeleton

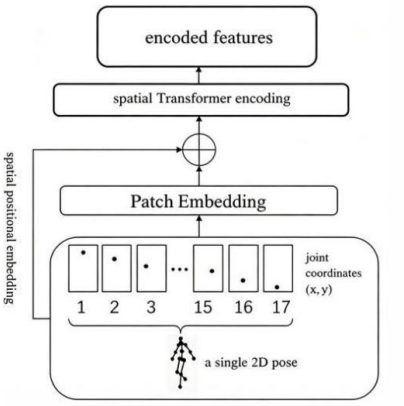

FIGURE 11. Spatial Transformer module.

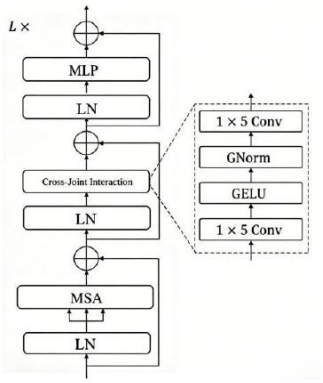

FIGURE 12. Spatial Transformer encoder with cross-joint interaction.

sequence $x_i \in \mathbb{R}^{1\times(j\cdot 2)}$ of frame i, 2 represents the two-dimensional coordinates of each joint, and J represents the number of joints in the human body.

The joint coordinates are mapped to spatial patch embeddings through a trainable linear projection layer, and spatial positional embeddings $E_{\mathrm{SPos}} \in \mathbb{R}^{J\times c}$ are added to obtain the fused feature sequence $Z_{0i} \in \mathbb{R}^{J\times c}$. The calculation is given in Eq. (20):

$$Z_{0i} = [x_{i1}E; x_{i2}E; \ldots; x_{ij}E] + E_{\mathrm{SPos}} \tag{20}$$

Here, x_{ij} denotes the 2D coordinate of joint j in frame i; $E \in \mathbb{R}^{(J\cdot 2)\times c}$ is a learnable linear projection matrix; and c is the spatial-embedding dimension. After passing through L encoding layers, the resulting space-encoded feature representation is denoted as $Z_{Li} \in \mathbb{R}^{J\times C}$.

Because the nonlocal nature of multi-head self-attention (MSA) may overlook meaningful low-score relationships, a cross-joint interaction module is proposed and integrated into the encoder [17]. The improved encoder architecture is shown in Fig. 12. The module consists of two 1×5 convolution layers, one Group Normalization (GN) layer, and a GELU activation function. Layer normalization (LN) and residual connections are used, and the encoder-block equations are as follows:

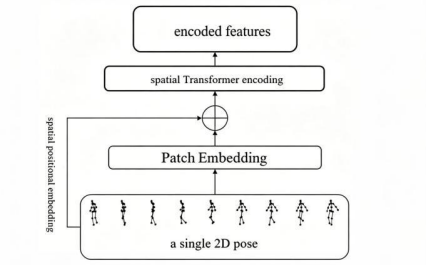

FIGURE 13. Temporal Transformer module.

$$Z'_l = \mathrm{MSA}(\mathrm{LN}(Z_{l-1})) + Z_{l-1}, \quad l = 1, 2, \ldots, L, \tag{21}$$

$$Z'_l = \mathrm{CONV}(\mathrm{GN}(\mathrm{GELU}(\mathrm{CONV}(\mathrm{LN}(Z'_l))))) + Z'_l, \quad l = 1, 2, \ldots, L, \tag{22}$$

$$Z_l = \mathrm{MLP}(\mathrm{LN}(Z'_l)) + Z'_l, \quad l = 1, 2, \ldots, L, \tag{23}$$

$$Y = \mathrm{LN}(Z_L). \tag{24}$$

The temporal Transformer module models inter-frame temporal dependencies, and its structure is shown in Fig. 13.

The input is the feature sequence $\{Z^i \in \mathbb{R}^{1\times(J\cdot c)} \mid i = 1, 2, \ldots, f\}$ output by the spatial Transformer. After transformation by a linear projection matrix $E_t \in \mathbb{R}^{(J\cdot c)\times C}$, temporal positional embeddings E_{TPos} are added, where C is the temporal-embedding dimension. The linearly projected fused feature sequence is calculated as shown in Eq. (25):

$$Z_0 = [z_1E_t; z_2E_t; \ldots; z_fE_t] + E_{\mathrm{TPos}} \tag{25}$$

Once the fused features pass through L Transformer encoder layers, the output encoded features $Z_L \in \mathbb{R}^{f\times C}$ are obtained.

A conventional temporal Transformer does not adequately model inter-frame temporal continuity for the same joint and may suppress features of occluded joints [19]. A cross-frame interaction module is therefore integrated into the encoder, using bilinear pooling to learn inter-frame joint interactions. The improved architecture is shown in Fig. 14 [20].

The cross-frame interaction module uses convolution layers to extract query Q, key K, and value V from the input fused feature Z as shown in Eq. (26):

$$Q = ZW_Q, \quad K = ZW_K, \quad V = ZW_V \tag{26}$$

where W_Q, W_K and W_V are learnable linear transformations. Bilinear matrix multiplication among Q, K, and V is defined in Eq. (27):

$$C = K \otimes Q, \quad Z = C \otimes V \tag{27}$$

where $C \in \mathbb{R}^{f\times f}$ denotes the cross-frame interaction matrix; $Z_{\mathrm{out}} \in \mathbb{R}^{f\times c}$ denotes the output feature; and $\otimes$ denotes bilinear pooling.

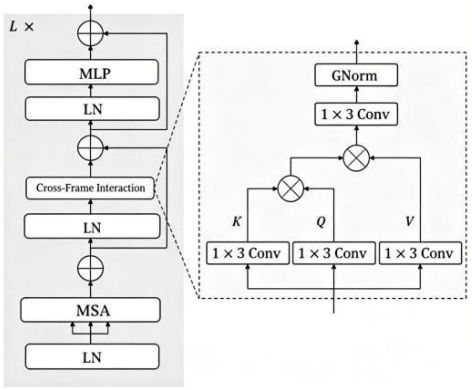

FIGURE 14. Temporal Transformer encoder with cross-frame interaction.

As in the other modules, layer normalization (LN) is applied before the cross-frame interaction module, followed by a residual connection. The final equations of the temporal Transformer encoder block are as follows:

$$Z'_l = \mathrm{MSA}(\mathrm{LN}(Z_{l-1})) + Z_{l-1}, \qquad l = 1, 2, \ldots, L, \tag{28}$$

$$Q = Z'_l W_Q, \quad K = Z'_l W_K, \quad V = Z'_l W_V, \qquad l = 1, 2, \ldots, L, \tag{29}$$

$$Z'_l = \mathrm{GN}(\mathrm{CONV}((K \otimes Q) \otimes V)) + Z'_l, \qquad l = 1, 2, \ldots, L, \tag{30}$$

$$Z_l = \mathrm{MLP}(\mathrm{LN}(Z'_l)) + Z'_l, \tag{31}$$

$$Y = \mathrm{LN}(Z'_l). \tag{32}$$

IV. EXPERIMENTAL RESULTS

A. DATASET AND EVALUATION METRICS

The experiments use the Human3.6M benchmark dataset for 3D human pose estimation [21], which contains approximately 3.6 million video frames with 3D pose annotations for 15 actions performed by 11 actors.

The division of the training, validation, and test sets is shown in Table 1.

Following the standard protocol, data from subjects S1, S5, S6, S7, and S8 are used for training, while data from S9 and S11 are used for testing.

Two common evaluation metrics for 3D human pose estimation are used: Mean Per Joint Position Error (MPJPE) and Procrustes-aligned Mean Per Joint Position Error (P-MPJPE) [22]. MPJPE (Protocol 1) is the mean Euclidean distance between the estimated and ground-truth 3D joint positions, with lower values indicating better performance. P-MPJPE (Protocol 2) first applies rigid alignment through Procrustes analysis to eliminate the effect of global transformations and then calculates MPJPE.

B. DATASET AND EVALUATION METRICS

ISTFormer is trained and validated on the Human3.6M dataset. It is implemented in PyTorch and accelerated using an NVIDIA GeForce GTX 3060 (12 GB) GPU. The Adam optimizer is used for 100 training epochs, with a weight decay of 0.1, an initial learning rate of 0.0001, exponential decay by a factor of 0.99 after each epoch, and a batch size of 512.

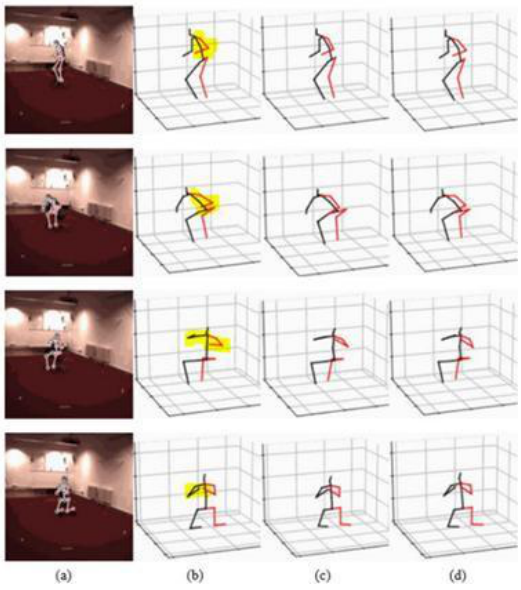

FIGURE 15. Visual comparison of 3D pose-estimation results: (a) input image; (b) ground truth; (c) PoseFormer; (d) ISTFormer.

The experiments use input-sequence lengths of 9, 27, and 81 frames. A Convolutional Pose Machine (CPN) generates the input 2D keypoints [23], and horizontal flipping is applied for data augmentation.

The spatial and temporal Transformer encoders each contain four layers, and the multi-head self-attention mechanism uses eight attention heads. The spatial-embedding dimension c is set to 32. CPN outputs 17 2D keypoints per frame, and the temporal-embedding dimension C is calculated as $32 \times 17 = 544$.

C. QUALITATIVE ANALYSIS

To qualitatively evaluate the 3D pose-estimation performance of ISTFormer, the estimated results are visualized and compared with the ground truth and PoseFormer, as shown in Fig. 15. For the "Sitting Down" action sequence of subject S11 in the Human3.6M test set, the yellow-marked regions show that ISTFormer's local joint-position estimates are closer to the ground truth and are therefore more accurate.

D. QUALITATIVE ANALYSIS

By comparing ISTFormer with PoseForme, StrideTransformer [24], P-STMO-S, GLA-GCN, DC-GCT [25], ConvFormer, and other state-of-the-art methods. All methods are trained and evaluated using detected 2D poses as input [26], and the test results are shown in Tables 2 and 3.

ISTFormer achieves the best average performance under both Protocol 1 and Protocol 2, outperforming the second-best method by 0.9% and 0.6%, respectively. Under Protocol 1, ISTFormer obtains the same mean error as StrideTransformer and ConvFormer using fewer input frames and is therefore more efficient. Under Protocol 2, it achieves the

TABLE 1. The Human3.6M Dataset

Group	Static	Source	Rate	Use
Upper-body actions	Directions	83856	50808	114080
	Discussion	154392	68640	140764
	Greeting	69984	33096	84980
	Posing	70948	25800	85912
	Shopping	49096	33268	48496
Full-body actions	Taking photos	67152	38216	89608
	Waiting	98232	54928	123432
	Walking	114468	47540	93320
Walking-related actions	Walking a dog	77068	30648	59032
	Walking together	76620	36876	52724
Chair-seated actions	Eating	109360	39372	97192
	Making a phone call	132612	39308	92036
	Sitting still	110228	46520	89616
	Smoking	138028	50776	85520
Ground-seated actions	Sitting down	112172	50384	105396
Other actions	Other	105576	–	–

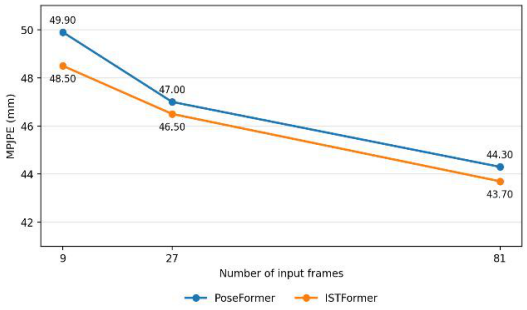

FIGURE 16. MPJPE of ISTFormer and PoseFormer with different input-sequence lengths.

best or second-best results for 12 of the 15 actions and the best performance for seven actions. Compared with the pretrained PoseFormer, it improves performance by 1.4% and 0.9% under the two protocols, respectively, verifying the effectiveness of the cross-joint and cross-frame interaction modules.

To evaluate the effect of input-sequence length, ISTFormer and PoseFormer are compared under Protocol 1, as shown in Fig. 16. As the input-sequence length increases, ISTFormer's estimation accuracy gradually improves, indicating strong adaptability to actions at different temporal scales [27]. ISTFormer can capture rapid joint movements and subtle changes over short time intervals, as well as long-term motion continuity and trends. It achieves lower MPJPE values than PoseFormer at all three sequence lengths, confirming that the proposed cross-joint and cross-frame interaction modules improve 3D pose estimation.

The core algorithms are quantitatively validated on the Human3.6M benchmark dataset. For the engineering application of power-grid pole-climbing operations, this section evaluates the scenario adaptability of the proposed framework from two perspectives: the generality of its methodological capabilities and qualitative testing in real scenarios.

The two core improvements of the proposed method address general spatio-temporal patterns of human motion and do not depend on a specific action category; they can therefore be transferred directly to pole-climbing scenarios:

The multi-scale temporal modeling capability of SMT-GCA can simultaneously capture motion patterns at different time scales during pole climbing, including short-term hand gripping and exertion, medium-term alternating limb movements, and long-term rhythmic adjustment of the center of gravity. The CBAM attention mechanism can adaptively strengthen the feature weights of discriminative joints such as the hands, feet, and trunk and suppress noise caused by tower backgrounds and tool occlusion, making it highly compatible with the action characteristics of pole climbing.

ISTFormer's cross-joint and cross-frame interaction modeling essentially uses human-skeleton topology constraints and temporal continuity to resolve depth ambiguity. For limb-crossing self-occlusion and depth-direction posture changes commonly encountered during pole climbing, inter-frame motion-consistency constraints can improve the stability of 3D reconstruction. This capability is broadly applicable across scenarios.

Publicly available inspection videos of power-grid pole-climbing operations are selected for qualitative validation. The test samples cover normal climbing, tool transfer, and posture adjustment, representing three typical operation types. The video resolution is 1920×1080 at 25 fps. The results show that:

During 2D keypoint detection, OpenPose stably extracts the coordinates of 17 core human joints under strong outdoor illumination and complex tower backgrounds. No obvious keypoint drift occurs in limb-crossing occlusion scenes, providing reliable skeleton input for subsequent modules.

During action recognition, SMT-GCA accurately distinguishes the three action states of climbing, pausing, and tool operation. It stably recognizes periodic motion patterns with alternating limbs and can support temporal change detection in the operation process.

During 3D pose reconstruction, the 3D skeleton output by

TABLE 2. Comparison of various methods under Protocol 1

Method	Directions	Discussion	Greeting	Conversation	Shopping	Photo	Waiting	Walking	Walk dog	Walk together	Eating	Phone call	Sitting	Smoking	Sitting down	Average
PoseFormer (no PT), $f = 81$	43.8	47.9	45.5	44.3	45.8	55.7	45.4	32.5	48.6	33.8	43.8	49.7	57.7	47.4	66.3	47.2
PoseFormer (PT), $f = 81$	41.5	44.8	42.5	42.1	42.0	51.6	43.3	31.8	46.1	32.2	39.8	46.5	53.3	45.5	60.7	44.3
StrideTransformer, $f = 351$	40.3	43.3	42.3	41.8	40.5	52.3	43.0	30.0	44.2	30.2	40.2	45.6	55.9	44.2	60.6	43.7
P-STMO-S, $f = 81$	41.7	44.5	42.9	42.8	41.3	51.3	42.8	30.8	43.8	30.7	41.0	46.0	54.0	45.0	61.0	44.1
GLA-GCN, $f = 243$	41.3	44.3	41.8	42.1	41.5	54.1	42.8	29.4	45.9	29.9	40.8	45.9	57.8	45.0	62.9	44.4
DC-GCT, $f = 1$	42.2	47.3	47.6	45.3	43.8	56.1	44.7	36.8	51.0	38.3	44.6	49.7	55.3	48.5	59.4	47.4
ConvFormer, $f = 143$	41.8	43.6	43.2	42.7	41.2	52.8	41.9	29.7	44.7	31.1	39.3	44.9	53.1	45.0	60.9	43.7
ISTFormer, $f = 81$	40.7	44.1	41.5	41.2	40.8	52.8	41.8	29.2	44.7	31.1	39.3	44.9	53.1	45.0	60.9	43.7

TABLE 3. Comparison of various methods under Protocol 1

Method	Directions	Discussion	Greeting	Conversation	Shopping	Photo	Waiting	Walking	Walk dog	Walk together	Eating	Phone call	Sitting	Smoking	Sitting down	Average
PoseFormer (no PT), $f = 81$	33.6	37.1	36.7	33.9	34.7	42.2	34.3	25.3	37.6	27.8	35.4	37.8	47.0	38.2	53.4	37.0
PoseFormer (PT), $f = 81$	32.5	34.8	34.6	32.1	32.0	39.5	32.4	24.5	35.3	26.0	32.6	35.3	42.8	34.8	48.5	34.6
StrideTransformer, $f = 351$	32.7	35.5	35.4	33.0	31.9	41.6	33.5	23.9	35.1	25.0	32.5	35.9	45.1	36.3	50.1	35.2
GLA-GCN, $f = 243$	32.4	35.3	34.2	32.1	31.9	42.1	32.4	23.5	35.6	24.7	32.6	35.0	45.5	36.1	49.5	34.8
DC-GCT, $f = 1$	34.7	37.8	39.8	36.2	34.2	44.1	34.9	29.5	41.1	32.5	35.4	39.4	44.5	39.9	48.6	38.2
ConvFormer, $f = 143$	31.9	34.4	35.0	32.9	31.8	40.7	31.5	23.6	–	–	–	–	–	–	–	–
ISTFormer, $f = 81$	31.4	34.6	33.7	31.6	31.0	39.7	31.3	23.4	–	–	–	–	–	–	–	–

ISTFormer satisfies human-kinematic constraints. The spatial relationships between the hands and feet and the pitch angle of the trunk during climbing are consistent with the actual movement logic. No obvious depth distortion or joint misalignment is observed, enabling intuitive reconstruction of the worker's 3D motion pose.

Because of the limited availability of professional motion-capture equipment, high-accuracy 3D ground-truth annotations for pole-climbing actions cannot currently be obtained, and quantitative error metrics cannot be reported. Nevertheless, the qualitative results verify the feasibility and adaptability of the proposed method in real pole-climbing operation scenarios and provide a technical basis for subsequent industrial deployment.

V. CONCLUSION

To address two key challenges in skeleton-based human action recognition-namely, insufficient modeling of multi-scale motion patterns and depth ambiguity during 2D-to-3D pose lifting-this study proposes an end-to-end 3D skeleton-based action-recognition framework for high-accuracy, real-time detection of changes in pole-climbing movements. The framework is important for preventing accidents in high-risk power-grid maintenance, communication-tower inspection, and professional sports training.

First, a Spatial Multi-scale Temporal Graph Convolutional Attention (SMT-GCA) network is designed. It uses parallel multi-branch dilated temporal convolutions to accommodate actions of different durations and integrates a convolutional attention module to improve the learning of discriminative features from key joints. An Interactive Spatio-Temporal Transformer (ISTFormer) architecture with dedicated cross-joint and cross-frame interaction modules is also developed, effectively improving spatio-temporal feature fusion and 3D pose-reconstruction accuracy under different lighting conditions and partial occlusion. On this basis, an integrated end-to-end web-based system is constructed for on-site deployment, supporting the complete workflow from RGB video input to visualized action-recognition results.

Extensive experiments on the Human3.6M benchmark show that the proposed method significantly and consistently outperforms state-of-the-art baseline models in both 3D pose estimation and action recognition, particularly in capturing fine-grained motion changes in complex outdoor scenes with occlusion. Future work will focus on two directions. First, the framework will be extended to multi-person interaction scenarios, with model lightweighting for edge deployment and multimodal fusion explored to further improve robustness. Second, a dedicated action dataset will be collected for typical industrial scenarios such as power-grid pole-climbing operations, enabling scenario transfer and field validation of the proposed method and promoting its application in industrial safety monitoring and standardized action training.

FUNDING

Science and Technology Project Funding from State Grid Sichuan Electric Power Company (Project Number: B3192225Z000)

REFERENCES

[1] H. Wang, M. Chen, X. Xiong, et al., "Behavior Recognition Technology of Power Operator Based on OpenPose and AT-STGCN," Journal of Sichuan University of Science & Engineering (Natural Science Edition), vol. 36, no. 04, pp. 61–70, 2023.
[2] Z. Cao, G. Hidalgo, T. Simon, et al., "OpenPose: realtime multi-Person 2d pose estimation using part affinity fields," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 172–186, 2021.
[3] S. Jin, W. Liu, E. Xie, et al., "Differentiable hierarchical graph grouping for multi-person pose estimation," in Proceedings of the European Conference on Computer Vision, Glasgow, UK, 2020, pp. 718–734.
[4] G. Pavlakos, X. Zhou, K. G. Derpanis, et al., "Coarse-to-fine volumetric prediction for single-image 3D human pose," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7025–7034.
[5] D. Mehta, S. Sridhar, O. Sotnychenko, et al., "Vnect: real-time 3d human pose estimation with a single rgb camera," ACM Transactions on Graphics, vol. 36, no. 4, pp. 1–14, 2017.
[6] X. Sun, B. Xiao, F. Wei, et al., "Integral human pose regression," in European Conference on Computer Vision (ECCV), 2018, pp. 529–545.

[7] J. Martinez, R. Hossain, J. Romero, et al., "A simple yet effective baseline for 3d human pose estimation," in IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2659–2668.
[8] C.-H. Chen and D. Ramanan, "3d human pose estimation = 2d pose estimation + matching," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 7035–7043.
[9] H. Yang, H. Liu, Y. Zhang, et al., "Parallel Multi-scale Spatio-temporal Graph Convolutional Network for 3D Human Pose Estimation," Journal of Software, vol. 36, no. 5, pp. 2151–2166, 2024.
[10] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: convolutional 3d pose estimation from a single image," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2500–2509.
[11] C.-H. Chen, A. Tyagi, A. Agrawal, et al., "Unsupervised 3d pose estimation with geometric self-supervision," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5707–5717.
[12] B. X. Nie, P. Wei, and S. C. Zhu, "Monocular 3d human pose estimation by predicting depth on joints," in IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3447–3455.
[13] L. Zhao, X. Peng, Y. Tian, et al., "Semantic graph convolutional networks for 3d human pose regression," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3420–3430.
[14] D. Zhao and M. Zhi, "Spatial multiple-temporal graph convolutional neural network for human action recognition," Journal of Frontiers of Computer Science & Technology, vol. 17, no. 3, p. 719, 2023.
[15] J. Liu, G. Wang, K. Hu, et al., "Global context-aware attention LSTM networks for 3D action recognition," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3671–3680.
[16] Y. Wang, M. Bao, and X. Liu, "3D Human Pose Estimation Based on Transformer and Graph Convolutional Network," Chinese Journal of Sensors and Actuators, vol. 38, no. 9, pp. 1624–1630, 2025.
[17] R. Zhao, K. Wang, H. Su, et al., "Bayesian graph convolution LSTM for skeleton based action recognition," in IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 6881–6891.
[18] C. Si, W. Chen, W. Wang, et al., "An attention enhanced graph convolutional LSTM network for skeleton-based action recognition," in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 1227–1236.
[19] T. S. Kim and A. Reiter, "Interpretable 3D human action analysis with temporal convolutional networks," in IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 1623–1631.
[20] P. Wang, W. Li, C. Li, et al., "Action recognition based on joint trajectory maps with convolutional neural networks," Knowledge-Based Systems, vol. 158, pp. 43–53, 2018.
[21] F. Yang, Y. Wu, et al., "Make skeleton-based action recognition model smaller, faster and better," in ACM International Conference on Multimedia in Asia, 2019, pp. 1–6.
[22] L. Shi, Y. Zhang, J. Cheng, et al., "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 12026–12035.
[23] Z. Liu, H. Zhang, Z. Chen, et al., "Disentangling and unifying graph convolutions for skeleton-based action recognition," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 143–152.
[24] F. Ye, S. Pu, Q. Zhong, et al., "Dynamic GCN: context-enriched topology learning for skeleton-based action recognition," in ACM International Conference on Multimedia, 2020, pp. 55–63.
[25] W. Peng, J. Shi, and G. Zhao, "Spatial temporal graph deconvolutional network for skeleton-based human action recognition," IEEE signal processing letters, vol. 28, pp. 244–248, 2021.
[26] Z. Chen, S. Li, B. Yang, et al., "Multi-scale spatial temporal graph convolutional network for skeleton-based action recognition," Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 2, pp. 1113–1122, 2021.
[27] J. Lee, M. Lee, D. Lee, et al., "Hierarchically decomposed graph convolutional networks for skeleton-based action recognition," in IEEE/CVF International Conference on Computer Vision (ICCV), 2023, pp. 10444–10453.