ITC 4/53

Information Technology and Control
Vol. 53 / No. 4/ 2024
pp. 1042-1059
DOI 10.5755/j01.itc.53.4.37243

A Study on 3D Human Pose Estimation with a Hybrid Algorithm of Spatio-temporal Semantic Graph Attention and Deep Learning

Received 2024/05/10 | Accepted after revision 2024/07/18

HOW TO CITE: Lin, S. (2024). A Study on 3D Human Pose Estimation with a Hybrid Algorithm of Spatio-temporal Semantic Graph Attention and Deep Learning. *Information Technology and Control*, 53(4), 1042-1059. https://doi.org/10.5755/j01.itc.53.4.37243

A Study on 3D Human Pose Estimation with a Hybrid Algorithm of Spatio-temporal Semantic Graph Attention and Deep Learning

Shengqing Lin
School of Computing and Information Sciences, Fuzhou Institute of Technology, Fuzhou, Fujian 350506, China

Corresponding author: lsq@fit.edu.cn

This paper introduces a method to enhance 3D human pose estimation accuracy by leveraging human topological structure and temporal information, addressing inaccuracies due to occlusion and complex poses. It proposes a spatiotemporal Transformer network that aggregates local temporal information to predict 3D poses for video frames, reducing sequence length through cross-step convolution. To further handle occlusion and information loss, the paper suggests a spatiotemporal graph attention network that incorporates spatial constraints and graph convolution with an improved adjacency matrix to emphasize local information in pose inference. A temporal convolutional network is also employed to model time, and the network alternates between temporal and spatial attention modules to prevent spatiotemporal information loss. Experiments on Human3.6m and HumanEva datasets demonstrate that the proposed method outperforms other approaches in prediction accuracy.

KEYWORDS: 3D human pose estimation; Graph Convolutional Neural Network; Self attention; Transformer.

1. Introduction

3D human pose recognition has important application value in human pose recognition, human-computer interaction, virtual reality, and action analysis. Traditional pose estimation methods rely on artificially set characteristics, and the model construction process is relatively cumbersome, making it difficult to meet the needs of complex environments such as complex scenes and rapid movements. In recent years, the rapid development of Convolutional Neural Network (CNN) technology has gradually shifted research on 3D human pose recognition towards deep learning. Li et al. (2014) first proposed a 3D human pose modeling method based on deep convolutional neural networks. This method uses a multitask fusion approach to jointly learn the attitude regression model and component detectors, and estimates the target through a regression network. Compared to directly inverting the three-dimensional pose, this method can better preserve the information in the image [1]. Zhou proposed a method for predicting the three-dimensional pose of robots by constructing a three-dimensional model of "parts heart heat" to construct a volume in space and using ensemble methods for end-to-end learning [2].

Zheng introduced a deep convolutional autoencoder (MR-DCAE) model based on stream regularisation for unauthorised broadcast identification. A specially designed autoencoder (AE) is optimised by entropic stochastic gradient descent, and then the reconstruction error in the testing phase is used to determine whether the received signal is authorised or not. To make this metric more discriminative, a similarity estimator across different dimensional manifolds is designed as a penalty term to ensure their invariance during gradient backpropagation [3]. Tang proposed a new spatio-temporal interactive attention model to address this issue. The method spatiotemporally encodes the input features and divides them into two equivalent parts, taking into account both temporal and spatial effects. In order to investigate a multi-node interaction model based on visual perception, multiple superconducting elements are stacked using multiple superconducting elements and a novel structure-enhanced local embedding (SPE) method is introduced in STCFormer. The method consists of a spatio-temporal convolution operation and a position-based embedding method for obtaining local information and describing the region where each node is located. To solve this problem [4], Zheng proposed to accomplish real-time AMC by constructing MobileViT, a lightweight neural network driven by clustered constellation images.Firstly, clustered constellation images were converted from I/Q sequences to help extract robust discriminative features. Then, a lightweight neural network called MobileViT was developed for real-time constellation image classification. Experimental results using an edge computing platform on the publicly available dataset RadioML 2016.10a demonstrate the superiority and efficiency of MobileViT. In addition, extensive ablation tests demonstrate the robustness of the proposed method to learning rate and batch size [5]. Zheng proposed a real-time AMC method based on a lightweight Mobile Radio Transformer (MobileRaT). The constructed radio transformer is iteratively trained while pruning the redundant weights according to the information entropy, so that robust modulation knowledge can be learnt from multimodal signal representations to perform the AMC task. This is the first attempt to integrate and apply pruning techniques and lightweight transformer models to process timing signals, thereby improving their inference efficiency while ensuring AMC accuracy. Finally, the experimental results validate the superiority of MobileRaT by comparing it with a series of state-of-the-art methods based on two public datasets. When processing RadioML 2018.01A and RadioML 2016.10A, the two models MobileRaT-A and MobileRaT-B achieve an average AMC accuracy of 65.9% and 62.3%, respectively, and the highest AMC accuracies of 98.4% and 99.2% at +18 dB and +14 dB, respectively [6]. Jiang proposed a zero-sample diffusion optimisation (ZeDO) pipeline based on 3D HPEs to solve the problem of cross-domain and field 3D HPEs. Multi-hypothesis ZeDO achieves state-of-the-art (SOTA) performance of minMPJPE 51.4mm on Human3.6M without the need to use any 2D-3D or image 3D pairs for training. In addition, the single hypothesis ZeDO achieves SOTA performance on the 3DPW dataset with PA-MPJPE 42.6mm in a cross-dataset evaluation, which outperforms even learning-based methods trained on 3DPW [7]. Li combines the estimation of 2D target pose with object recognition and infers the dynamic pose of the target by using a recursive multi-layer Transformer network [8].

Jiang reviews the results of the fast-growing research on the use of different graph-based deep learning models (e.g., Graph Convolutional Networks and Graph Attention Networks) in a variety of problems in different types of communication networks (e.g., wireless networks, wired networks, and software-defined networks). We also provide a well-organised list of problems and solutions for each study and point out future research directions [9].

Traffic prediction is important for the success of intelligent transport systems. Deep learning models including convolutional neural networks and recurrent neural networks have been widely used in traffic prediction problems to model spatial and temporal dependencies. In recent years, graph neural networks have been introduced in order to model graph structures as well as contextual information in traffic systems and have achieved state-of-the-art performance in a range of traffic prediction problems. Jiang reviewed the rapidly growing body of research results on the use of different graph neural networks (e.g., graph convolutional networks and graph attention networks) for a wide range of traffic prediction problems, such as road traffic flow and speed prediction, passenger flow prediction in urban rail systems, and demand prediction in ride-hailing platforms. This study also provides a comprehensive list of open data and source code for each problem and identify future research directions. Recent research results show that deep learning techniques have made significant progress in the field of 3D human pose estimation [10]. From Li's direct prediction method to Zhou's part-centre-heatmap triad, to Liu's GAST-Net, Tang's STC block, Aksan's self-attentive architecture, and Zhang's spatial graph feature acquisition scheme, each study builds on the previous ones to further improving the accuracy and robustness of pose estimation. These methods show their respective advantages in dealing with complex scenes, fast motion, depth ambiguity, and self-obscuration problems, and provide a variety of effective solutions for 3D human pose estimation.

This paper is based on a 2D skeleton model and conducts research in three aspects: the 3D-SLAM algorithm for monocular visual images; Establish a 3D DAR model; Use graph convolution and attention theory to construct a 3D DAR model. On this basis, this paper proposes a video spatiotemporal fusion method based on the spatiotemporal domain transformation structure, which combines local time-domain information with a gradually reduced method to predict the 3D human pose in intermediate frames. The research content includes the following aspects: establishing spatiotemporal correlation models based on attention mechanisms; Establish a spatiotemporal correlation model based on time-domain convolutional networks; A spatiotemporal correlation model based on visual perception to solve occlusion problems [11-16].

This paper focuses on the research of 3D human posture methods based on 2D skeleton, and the Main contributions are:

1. Predicting 3D human posture using monocular pictures, using a priori information of human posture topology map, combining graph convolution and attention to fuse local and global posture information to improve the prediction accuracy.
2. Extracting spatio-temporal information from the video using the spatio-temporal Transformer structure, aggregating local temporal information and gradually reducing the sequence length to predict the 3D human pose in the centre frame.
3. Modelling local and global spatial information by attention mechanism, extracting temporal information by temporal convolutional network, and designing a network structure interleaving spatial semantics and temporal dependence to alleviate the occlusion problem.

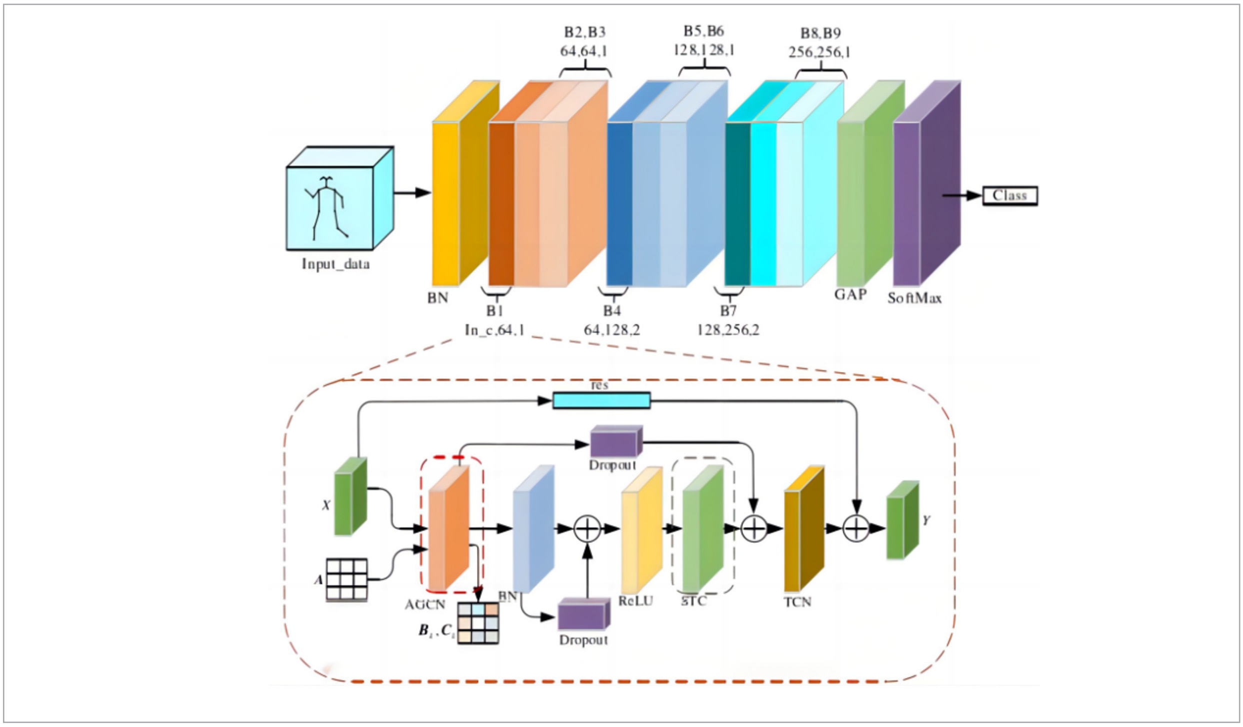
2. Related Work

To solve the problems of traditional spatio-temporal graph convolutional networks, this section improves the spatial graph convolutional GCN in ST-GCN and proposes adaptive data-driven graph convolution, which can flexibly change the topology structure of the skeleton graph based on the data samples learned during training, thus adapting to different action types and data characteristics [17]. A multi-dimensional attention mechanism was designed using attention mechanism to guide the model to focus on the main features and reduce the impact of redundant features. This section constructs an action recognition network based on data-driven graph convolution and attention mechanism, and the overall model structure is shown in Figure 1. The red and green dashed boxes represent the adaptive data-driven graph convolution and multi-dimensional attention mechanism modules, respectively, using ST-GCN's temporal convolutional network TCN in the temporal dimension.

Using a single spatio-temporal graph convolution module as one basic unit, the entire action recognition network consists of 9 spatio-temporal graph convolution units, as shown in Figures B1 to B9 in Figure 1, B1 to B9 are divided into three groups based on the number of channels, with each group having 64, 128,

Figure 1

Overall Structure of Action Recognition Network Model



and 256 channels, respectively. The model adds a batch normalization layer in the input stage to normalize the data.

At the end, global average pooling is used to transform the sample feature map to a uniform size. Finally, Soft Max is used for predicting and classifying action features. In order to stabilize the model training and avoid gradient explosion or vanishing, residual connections are added to the module. After passing through the residual module, the input is added to the output of the temporal convolutional network TCN as the overall output of the module.

2.1. Adaptive Data-driven Graph Convolution

To solve the problem of fixed topology in the traditional spatio-temporal graph convolutional network ST-GCN, this paper replaces the original predefined adjacency matrix with an adaptive adjacency matrix driven by data samples. Adaptive adjacency matrix can adaptively optimize the topology of the graph during the training process, modify the adjacency matrix features of nodes, and share the topology structure of the graph in multi-layer graph convolution. To ensure the stability of the original model, residual structures are added to connect different branches. The proposed adaptive data-driven graph convolution AGCN is shown in Figure 2.

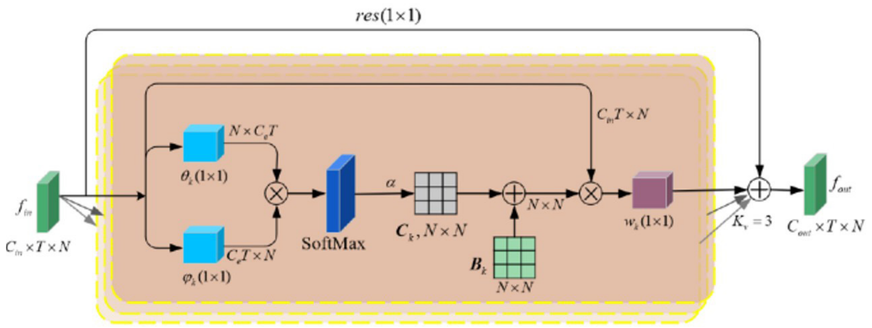
In ST-GCN, the adjacency matrix A_k actually determines the topological form of the graph, while the mask matrix B_k represents the connection strength between the root node and its neighboring nodes. This article improves Ak and Bk to achieve adaptive topology of the graph structure. The ACGN theoretical calculation method in Figure 2 is shown in Equation (1):

$$f_{out} = \sum_{k}^{K_v} W_k f_{in}(B_k + \alpha C_k) \cdot \tag{1}$$

Equation (1) introduces two different matrices based on the adjacency matrix A_k, namely B_k and C_k, using B_k instead of the mask matrix M_k and A_k. B_k initializes using A_k in the formula. α is a parameterized coefficient that is learned and updated through data samples during the training process.

Figure 2

Adaptive data-driven graph convolutional AGCN structure diagram



Among them, B_k has the same size as the adjacency matrix A_k, which is a data-driven global adaptive graph adjacency matrix with the same initial parameters as the adjacency matrix A_k. The elements of adjacency matrix B_k can be optimized by the optimizer like other elements and are not restricted. Through this data-driven adaptive graph structure, the network model can continuously optimize its graph convolution parameters to adapt to different action recognition tasks. In the original formula, the function of M_k is to provide different levels of attention to different adjacent nodes. However, in M_k, it is not possible to establish edges that do not naturally physically connect the human skeleton. In the adjacency matrix B_k, the size of its elements is continuously optimized through data-driven training, and the values of its elements are not limited. It can not only establish connections for physically nonadjacent nodes, but also represent the strength of the connections using the size of the values. Therefore, replacing the mask matrix M_k with the adjacency matrix B_k will make graph convolution more flexible. Adjacency matrix C_k, also known as similarity matrix, is a local graph related to data that expresses the similarity between two nodes in the graph. The normalized Gaussian function representation of the connection relationship and interaction strength between nodes v_i and v_j on the skeleton structure is shown in Equation (2):

$$f(v_i, v_j) = \frac{e^{\theta(v_i)^T \phi(v_j)}}{\sum_{j=1}^{N} e^{\theta(v_i)^T \phi(v_j)}}. \qquad (2)$$

In the formula, N is the total number of nodes on the skeleton.

The interaction connection between two nodes is calculated using the dot product of vectors. Let the channel size of the input feature f_{in} of the network model be C_m×T×N, and use embedding functions θ and ϕ to map the input feature map size to size C_e×T×N. For the embedding function, this article uses two 1x1 convolution kernels to implement the embedding functions θ and ϕ. The size of the input feature map is transformed, and the output feature maps of the two convolution kernels are N×C_eT and C_eT×N, respectively. The matrix multiplication is performed on the two output feature maps to eliminate the dimension C_eT and obtain a similar adjacency matrix C_k with a size N×N, where the values of the elements of the matrix represent the similarity of their corresponding nodes v_i and v_j.

The normalization effect of Gaussian embedding function in Equation (2) is similar to the structural effect of the SoftMax function. Therefore, to facilitate the implementation of above function using neural network layers, the SoftMax function can be used to calculate the similarity matrix C_k of nodes, as shown in Equation (3):

$$C_k = SoftMax(f_{in}^T W_{\theta k}^{TK} W_{\phi k} f_{in}). \qquad (3)$$

In the formula, W_θ and W_φ represent the parameters of the embedding functions θ and ϕ, respectively.

2.2. Multidimensional Attention Mechanism

The core principle of attention mechanism is to find the correlation between local data based on the input raw data, and to reallocate attention computing resources reasonably. In research based on skeleton key points, it is particularly important to find the correlation between two spatially nonadjacent joints. Taking inspiration from this, this article designs a multi-dimensional attention mechanism module. From the perspective of spatio-temporal channels, three modules were established: spatial feature attention enhancement module, temporal feature attention enhancement module, and channel feature attention enhancement module. Guide the model to focus on key spatiotemporal channel features and reduce the impact of redundant features. The output feature map of each module is multiplied by the input feature map to enhance attention based on the original feature map. The following will introduce the implementation methods of the three modules built in sequence.

Spatial feature attention enhncement module: This module provides different attention intensities to key points at different spatial angles, enhancing the feature expression of joint features related to actions during training and learning in the model. Its calculation is shown in Equation (4):

$$M_s = \sigma(g_s(AvgPool_t(x))) . \tag{4}$$

In the formula: σ is a nonlinear transformation function, which is completed using the Sigmoid activation function in this section. The curve of the Sigmoid function is an S-shaped growth curve that can map input variables between (0,1). The curve has the characteristics of smoothness and easy differentiation, which is more suitable for the data-driven adaptive graph convolutional network proposed in this paper and can strengthen the role of resource reallocation in attention mechanisms. The *s* in g_s represents spatial angle, and g_s represents feature extraction at spatial angle. In this section, one-dimensional convolution is used to generate spatial attention intensity for different key points at spatial angle, enhancing or weakening the influence of different joints in action features. $X \in R^{C\times T\times N}$ is the input feature map of the spatial feature attention enhancement module, with a number of channels of $C\times T\times N$. The average pooling layer *AvgPool* is used to take the mean of the input feature map on the spatial channels, and channel transformation is performed on the input feature map to transform its channel number from $C\times T\times N$ to $C\times 1\times N$, compressing the time dimension to fully enhance the spatial attention feature. $M_s \in R^{1\times 1\times N}$ is the output feature map of the spatial feature attention enhancement module, and the result of point multiplication with the input feature map is the addition of the input feature map and input to the next module.

Time feature attention enhancement module: The function of this module is similar to the SAM module, guiding the model to enhance its attention to local action time segments during training, while suppressing the influence of irrelevant time segments, improving the efficiency of training and learning. The calculation method is shown in Equation (5):

$$M_t = \sigma(g_t(AvgPool_s(X))), \tag{5}$$

where: $M_t \in R^{1\times T\times 1}$ is the output of the time feature attention enhancement module, which is input into the next module after matrix dot multiplication and addition with the output of the previous module; $AvgPool_s$ represents performing average pooling on the spatial dimension of the input feature map; g_t performs convolution on the pooled feature map in the time dimension, similar to the SAM module, which uses one-dimensional convolution to implement this function.

Channel feature attention enhancement module: The CAM module emphasizes that the data features of different channels have different semantic levels in the action category judgment of different data samples, enhancing the feature discrimination ability of channels. The calculation method of CAM is shown in Equation (6):

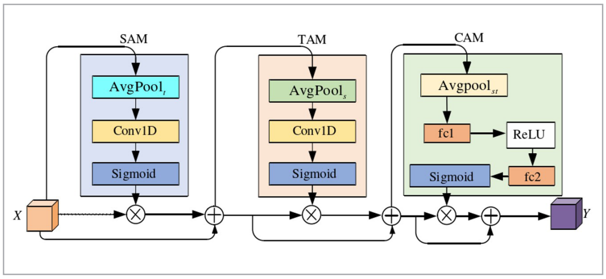
$$M_c = \sigma(g_{c2}(\delta(g_{c2}(AvgPool_{st}(X))))) . \tag{6}$$

In the formula, $M_c \in R^{C\times 1\times 1}$ represents the feature map output by the CAM module, which is then multiplied and added with the input of the module before being output; $AvgPool_{st}$ performs average pooling on the input feature map in both spatial and temporal dimensions and performs size transformation; The g_{c1} and g_{c2} of the CAM module are different from the one-dimensional convolution of the above modules. In the CAM module, g_{c1} and g_{c2} are linear functions implemented using fully connected layers, mainly used for feature extraction along the channel dimension; The δ function is implemented using the *ReLU* activation function.

Arrange the three modules in order of space, time, and channel, and input the feature maps into the three modules in sequence for attention feature enhancement. As shown in Figure 3, the three modules in the figure are combined into a SAM-TAM-CAM structure, or STC for short.

Figure 3

STC attention mechanism fusion module



The output of each module performs a point multiplication operation with its output, and then adds the input and point multiplication results to the next module, so as to achieve the fusion operation of the three attention module features. The input of the STC module is the output feature map of the graph convolutional neural network, and the output of the STC module is further input into the temporal convolutional network to extract temporal features [18].

3. Research Methodology

The frame level progressive aggregation spatio-temporal Transformer network constructed in this section, given a sequence $P=\{p_1,p_2,...,p_T\}$ for two-dimensional pose estimation, aims to reconstruct the three-dimensional pose $X\in R^{j\times 3}$ of the intermediate frames in the video, where $p_t\in R^{J\times 2}$ represents the two-dimensional coordinate position of the joint points at frame t. Regarding video frames, T represents the quantity, while J represents the number of connections. This network consists of two modules, namely STEP and STEP. The research content includes the following points: using the STEP method to encode joints and obtaining the association between joints through multi view attention method, obtaining joint representations containing spatial information; Convert the dimension of joint representation to pose representation, and use stepwise convolution method to fuse local temporal information based on this. Use segmented convolution method to fuse local temporal information to obtain pose descriptions of each node in the image; Finally, linear projection is used to extract the pose of intermediate frames [19].

3.1. Space Encoder (STE)

By using the spatial Transformer module, a high dimensional representation of the bit positions of a single frame can be made and given 2D joint coordinates. The individual joint coordinates are then mapped to the high dimensional space using a linear layer and they are superimposed on the learnable spatial position embedding $E_{SPos}\in R^{J\times c}$. The position encoding combines the joint spatial position information with the joint point representation, enabling the encoder to actively learn the joint position information. The input $p_t\in R^{J\times 2}$ of frame t becomes $Z_0^t\in R^{J\times 2}$, where J represents the spatial embedding dimension, and then the obtained feature sequence Z_0^t is input into the spatial transformer encoder. The encoder is composed of 1N cascaded structures, with two sub modules in each layer, namely multi head attention and a feedforward network.

3.1.1. Capturing Global Attitude Information

Attention can calculate the influence between joint points, capture global pose information, and jointly model information from different subspaces at different positions using multi head attention. Each head uses scaled column dot product attention in parallel, and finally concatenates multiple attention heads as outputs.

$$MultiHead(Q,K,V)=Concat(H_1,H_2,...,H_h)W_{out}, \tag{7}$$

$$H_i=Attention(Q_i,K_i,V_i), i\in[1,2,...,h], \tag{8}$$

wherein

$Attention(Q,K,V)=Soft\max(QK^T/\sqrt{d})V$, Q, K and V are created by multiplying the input feature matrix by three weight matrices $W_Q, W_K, W_V\in R^{c\times c}$:

$$Q=ZW_Q, K=ZW_K, V=ZW_V. \tag{9}$$

The query matrix Q, key matrix K, and value matrix V are obtained by multiplying $W_Q, W_K, W_V\in R^{c\times c}$ with the attitude representation matrix $Z\in R^{J\times c}$, where $Q,K,V\in R^{J\times c}$ and $Q\times K$ are used to calculate the degree of mutual influence between each joint point. This influence matrix is multiplied by the value matrix V. At the same time, a scaling factor is used to normalize the output global attention matrix, effectively preventing gradient vanishing or exploding problems. Finally, the obtained joint representation has global spatial attitude information.

3.1.2. Consolidate Joint Information (FFN)

After the joint feature vector is output from Multi head attention, it has already fused joint information from other positions. FFN is used to consolidate the joint's own representation information, only using fully connected layers without considering the influence of neighboring nodes and performing feature transformation on its own position representation. This operation not only aims to integrate joint information from different positions in space for each joint representation, but also to consolidate the joint's own information, rather than simply weighted averaging spatial information.

3.1.3. Complete Process of Spatial Information Extraction

Using $Z \in R^{J \times c}$ as input, describe the data processing process for each layer as follows:

$$Z_l^{'} = MSA(LN(Z_{l-1})) + Z_{l-1}, \tag{10}$$

$$Z_l = FFN(LN(XZ_l)) + Z_l^{'}, \tag{11}$$

$$Z_L = LN(Z_L). \tag{12}$$

Among them, *LN* represents LayerNorm. In order to maintain the stability of data features, the input features of the previous layer are processed using *LN*, which can reduce the changes in feature distribution during nonlinear changes, accelerate network training, and also play a certain role in preventing overfitting.

3.2. Time Encoder (TTE)

After encoding a single frame pose into high-dimensional space using STE, the dependency relationship between time series is modeled using TTE. For the *i-th* frame, the STE output $Z \in R^{J \times c}$ is transformed into a vector $R^{1 \times (J \cdot c)}$, and then these vectors from the *f* input frame are connected as $Z \in R^{f \times C}$. Among them, $C = J \cdot c$ adds learnable temporal position encoding $E_{TPos} \in R^{f \times C}$ to preserve the frame position information. For TTE, similar to the STE structure, N_2 identical hierarchical connections are used, and each layer uses an attention module to extract global temporal information from the long sequence. For the feature of redundant information in the time series, a convolutional feedforward network (CFFN) is used to capture local temporal information while fusing redundant information, allowing the network to focus its attention on the pose of the intermediate frame.

CFFN can reduce redundant information while extracting local temporal information. It uses step convolution to fuse the pose representations of adjacent frames, supplementing some information that cannot be obtained due to occlusion. The pose representation changes from the output of the multi head attention layer to $Z \in R^{f \times Cin}$, which will be used as input. One dimensional convolution uses a kernel size of *K*, a step factor of *S*, and the convolutional feedforward network can be written as:

$$CFFN_{S(t),C_{cout}}(z) = \sum_i^C \sum_k^K w_{C_{out},i,k} * Z_{S(t-\frac{K-1}{2}+k),i}. \tag{13}$$

TTE reduces the length of the time dimension layer by layer and combines adjacent pose representations into shorter sequence length representations. At the same time, it reduces redundant information in the time series, and finally outputs the feature representation $Y \in R^{1 \times Cout}$ $Y \in R^{1 \times Cout}$ of the intermediate target frame.

Finally, use the LayerNorm layer and a linear MLP block to regress $Y \in R^{1 \times C_{out}}$ and output $y \in R^{1 \times (J \cdot 3)}$, outputting the 3D pose of the intermediate frames.

3.3. Loss Function

Mean squared error loss "Optimality criterion used to minimize euclidean distance between the predicted 3D poses and labeled joints:" is given by:

$$L = \frac{1}{J} \sum_{k=1}^{J} \left\| p_k - \hat{p}_k \right\|_2. \tag{14}$$

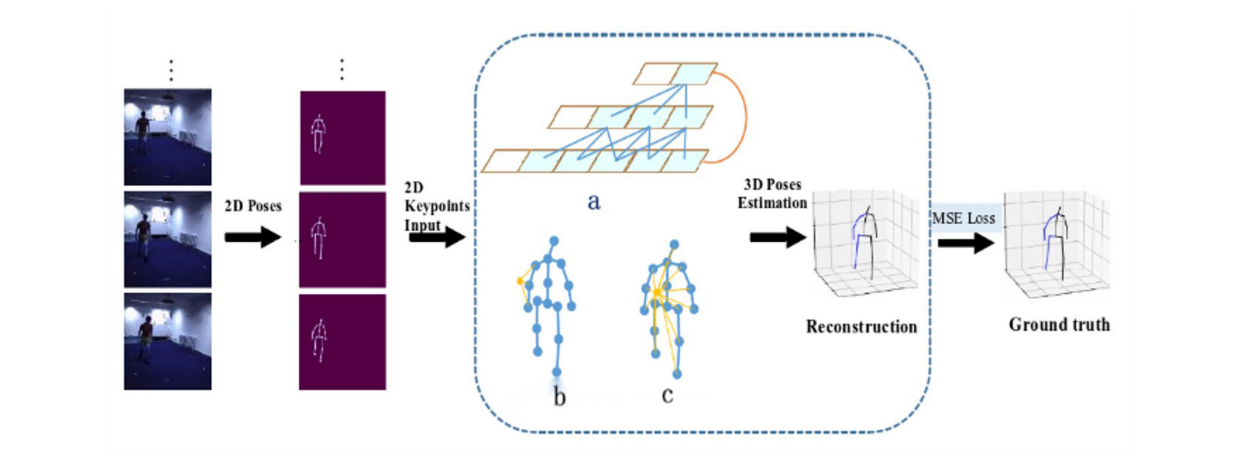
The values represented by p_k and $\hat{p}_k$ are actually the predicted locations of the *k-th* 3D corner point with respect to their true labels.

4. Experimental Setup

This network captures spatial information through graph attention blocks and models long-term contexts using temporal dilated convolution, as shown in Figure 4. The graph attention module learns the symmetry of skeleton joints, local kinematic relationships

Figure 4

Spatiotemporal Graph Attention Network



of joints, and global pose semantics, while TCN can flexibly capture changing time series information, as shown in Figure 4(a). For single frame scenes, dilated convolution can be replaced with stride convolution for fast inference without the need to retrain a new model. For time modeling, this section designs a TCN network and extends it to handle three-dimensional spatio-temporal sequences; For local spatial features, GCN is used to model local connections, symmetric connections, and kinematic connections, which are referred to as "local graph attention" in this section, as shown in Figure 4(b). For global spatial features, self-attention mechanism is used to express pose semantics through data-driven learning, which is called "global graph attention", as shown in Figure 4(c).

Meanwhile， Graph Attention was utilized to capture hierarchically structured human bodies， which could then be leveraged to obtain temporal-wide holistic meaning representation especially for local-space / Global-Space fusion as well as Interleave between Temporal Module & Spatio-Temporal Modules to extract and sift through spatiotemporal features of 2-dimensional joint sequences.

On this basis, a series of two-dimensional motion prediction models were proposed and simulated. The specific content involved includes based on TCN theory, researching time models based on TCN, and solving the problem of temporal data dependence; Research on skeleton motion and human symmetry modeling methods based on human spatial attention networks; Based on the global human spatial attention model, achieve effective representation of human spatial features.

4.1. Time Convolutional Network

The method consists of an input layer, an output layer and a B-layer time convolution module, which enables flexible tuning of the perceptual domain by adjusting the core size and convolution coefficients. One kind of 1D convolution with a core size of k and a null coefficient of $d=k^B$ is used for each piece, and then a convolution with a core size of 1 is used. On this basis, we replace the original 1D convolution with a 1D convolution with $k\times1$ core size. Based on the characteristics of the TCN network, we design a model that can vary over time within the perceptual field and transform it into two dimensions based on a one-dimensional BatchNorm, which is added to normalize the starting position of the network.

4.2. Local Attention Map

Within any given time frame, two-dimensional joints represent the joints of human skeletons, which can naturally be represented by undirected graphs, where joints are nodes and human limbs are edges. When using the SemGCN framework, the image can be modelled by constructing a skeletal graph of 2D nodes, representing the 2D bit-pose as a graph $g=(V,E)$, where N nodes are represented by sets of edges, and $X=\{x_1,x_2,...,x_N|x_1\in R^{1\times C}\}$ contains the set of node characteristics for the characteristics of C channels. The structure of the graph can be initialized with a first order continuous matrix $A \in R^{N \times N}$ which represents the links between nodes and a unitary matrix I which represents the self-connectivity. In GCN, $\tilde{A}=(A+I)$ is a convolution core. In SemGCN, given the features of l nodes, the following convolution operation is used to obtain the feature output of subsequent layers:

$$X^{(l+1)} = \rho(M \odot \tilde{A})X^{(l)}W \cdot \qquad (15)$$

Here there is a $W \in R^{C_l \times C_{l+1}}$ denoting a learnable matrix for the output channel transformation, and an $M \in R^{N \times N}$ denoting a mask matrix, and an $\odot$ denoting a symbol for cell-level multiplication, and a ρ denoting non-linear software for normalizing the effect of node characteristics on corresponding neighbouring nodes in the graph.

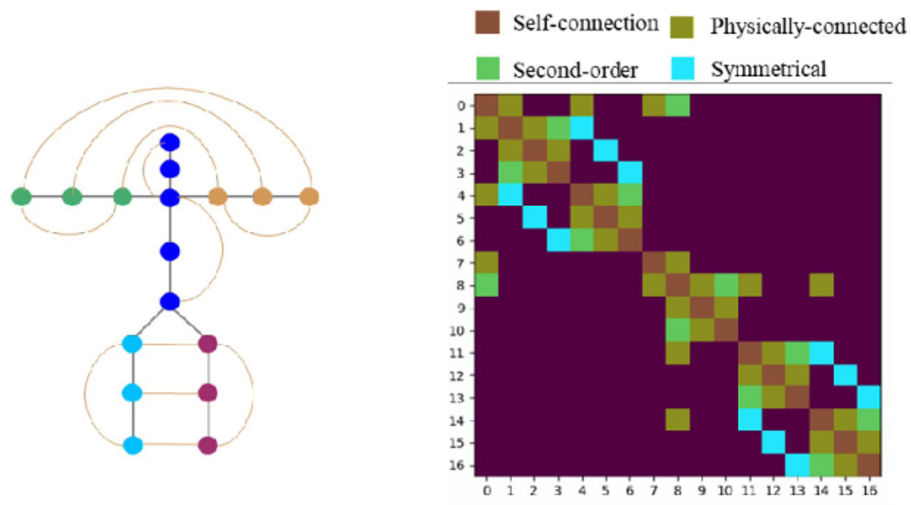
This formula can learn the spatial semantic information between adjacent nodes; however, the representation of first-order adjacent nodes is very poor for the symmetrical structure of the human body centered on the torso and the kinematic constraints in the human body. Therefore, this section explicitly considers structural knowledge related to human symmetry. In addition, the first-order model cannot accurately describe the spatial location of a person because it is limited to first-order neighbouring joints, i.e., the distal joints of the wrist, ankle, and head are only available at a more distant level and are not capable of effective spatial localization. Therefore, in order to mitigate the problem of inaccurate localization, second-order adjacent nodes (ankle-knee-hip), upper limbs (wrist-elbow-shoulder), head (head-neck-chest), and torso (chest-spine-pelvis) are used in this section [20-21].

To address these issues, this section adopts a convolution operator with a larger kernel to modify the classical GCN operation, grouping adjacent nodes based on semantics and using different kernels for different adjacent nodes. As shown in Figure 5(b), adjacent nodes are divided into four groups based on intuitive explanations:

1. Node itself A_{self}
2. Adjacent nodes on physical connections A_{phy}
3. Adjacent nodes indirectly "symmetrically correlated" A_{sym}.
4. Second-order adjacent nodes related to motion chains A_{sec}.

Figure 5

Human Skeleton Map and Its Grouping Adjacency Matrix



(a) Human Skeleton Map

(b) Group adjacency matrix

Based on the above classification, the update of graph convolution in Equation (16) becomes:

$$X^{(l+1)} = \sum\nolimits_k \rho(M_k \odot \tilde{A}) X^{(l)} W_k \tag{16}$$

where k represents the index of adjacent node types, $\tilde{A}$ and mask M_k are multiplied to obtain adjacency matrices for different classes, and W is the weight matrix of the k-th type of adjacent node.

3.3. Global Attention Map

The association between distal joints (e.g., wrist-ankle) is crucial for coding the overall body posture. This helps to solve problems such as distance ambiguity and occlusion between motion subs. In order to accommodate non-local relations and encode them efficiently, this paper plans to use a multi-attention end-to-end GCN to generalize the adjacency relations proposed in (17) to the whole.

$$\mathop{\|}_{k=1}^{K} (B_k + C_k) X^{(l)} W_k \,. \tag{17}$$

Among them, K is the number of attention heads, $B_k \in R^{N\times N}$ is the adaptive global adjacency matrix, $C_k \in R^{N\times N}$ is the learnable global adjacency matrix, and $W_k \in R^{C_l \times C_l}$ is the transformation matrix. In this experiment, 8 parallel attention heads were set up. Next, we will discuss in detail the design and role of the defined adaptive global adjacency matrix B_k and the learnable global adjacency matrix C_k.

B_k represents a data correlation matrix that learns the influence coefficients of all nodes on each node in the graph. In this section, the attention coefficient function is used to determine whether there are connections between nodes and the strength of the connections. That is to say, given two node features x_i and x_j, two functions θ and ϕ are first applied to down sample the features of each node from C_i to C_l/K channels. As the number of channels for each node decreases, the total computational cost of multi head attention is similar to that of single head attention for the entire channel. Then, the two vectors are dot products to calculate the correlation between the two nodes. In order to facilitate coefficient comparison between nodes, the output is normalized using the SoftMax function and scaled using d_k to avoid the gradient of the dot product being too small after passing through the SoftMax function. This process can be expressed by the following formula:

$$\alpha_{ij} = \frac{e^{\theta(x_i)\cdot\phi(x_j)/\sqrt{d_k}}}{\sum_{k=1}^{N} e^{\theta(x_i)\cdot\phi(x_k)/\sqrt{d_k}}}, \tag{18}$$

where θ and ϕ are linear functions; d_k is set to the number of channels through which the feature passes through the linear layer, C_l/K to prevent training instability caused by excessive weights. C_k is a learnable adjacency matrix with an initial value of 0. Unlike B_k, the value of C_k is updated by calculating the influence coefficient between each joint feature. The elements in C_k are arbitrary, indicating the correlation between the two joints learned by the network itself.

5. Experimental Results and Analysis

5.1. Training Set

In this section, we used four neuronal networks of different sizes, 9, 27, 81, and 243, to train the built model. For the two perceptual wilds of 9 and 27, we increased the number of output channels of the high-altitude convolutional neural network by 128 channels, respectively. While for the two perceptual fields of 81 and 243, we set 64 channels and 32 channels, respectively. Meanwhile, only the maximum maxima between the predicted 3-dimensional coordinates and the markers were calculated. In this section, the performance of Human 3.6 M and HumanEva-I is evaluated using the common evaluation metrics 1 and 2. The GPU used for the experiments was a Ge Force RTX 3080, Cuda version 12.0, and the model was running on an Ubuntu 16.04 system.

Training and inference settings: To train the model, a single frame prediction training strategy is used. When inferring, for predicting a single frame, using dilated convolution can waste a lot of computing power, reduce prediction efficiency, and increase prediction time. Therefore, it is necessary to replace dilated convolution with step convolution, which improves efficiency through layer-by-layer inference. When the input is the entire long video sequence, step convolution needs to be switched to dilated convolution, shifting from an optimized training strategy to layer by layer implementation to make faster predictions.

Experimental setup: The number of joints varies in different datasets. In Human3.6M, a total of 32 joints are labeled, and 17 of them are generally used to predict human posture. In HumanEva-I, 15 joints are used. In addition, high frame rates can lead to information redundancy, which can have a negative impact on the encoding of global semantics over time. For this reason, it is necessary to down sample the Human3.6M dataset from 50FPS to 10FPS. On the other hand, due to the short duration of videos in the HumanEva-I dataset, down sampling is not performed. In real-time estimation, the duration of long videos is not suitable for fast estimation, therefore, the model of 243 receptive field is not down sampled. Finally, in order to expand the dataset, this section uses horizontal flipping to enhance the data during training and testing.

This section uses the PyTorch deep learning framework to implement the constructed model and conducts end-to-end training. The Amsgrad optimizer is used for optimization, with a batch size of 128 and 60 epochs trained. The learning rate starts from 0.001, and then a learning decay factor of 0.95 is applied in each epoch, with the dropout rate set to 0.05 in each dropout layer. For HumanEva-I, the batch size is 32, the learning decay factor is 0.98, the dropout rate is 0.5, and 200 epochs are trained.

5.2. Ablation Experiment

In the ablation experiment, the two-dimensional pose detected by CPN was used as input in the model with a receptive field of 27. This section analyzes the impact of different spatial semantics on prediction results in network structures, as shown in Table 1. This section constructs a regular graph attention spatio-temporal network consisting of TCN and first-order SemGCN as the baseline, regresses two-dimensional joints to three-dimensional poses, and then adds different semantic GCNs for ablation research. The spatial semantics include local kinematic relationship $\hat{A}_{sec}$, symmetry relationship $\hat{A}_{sym}$, global adaptive matrix B_k, and global learnable matrix C_k. It can be seen that when additional local and global pose constraints are added, model performance steadily improves, with the greatest improvement coming from local kinematic connections, symmetry, and global adaptive matrices. These spatial constraints accurately express layered and symmetrical human structures and convey global pose semantics, proving the importance of rich spatial semantic information in 3D pose estimation tasks.

Table 1

Comparison of the Impact of Different Spatial Semantics

Method(T=27,CPN)	MPJPE(mm)	PMPJPE(mm)
Baseline	60.9	50.5
+Local GCNs with A_{sec}	53.6	41.5
+Local GCNs with A_{sym}	50.6	39.1
+Global GCNs with B_k	46.1	36.2
+Global GCNs with C_k	45.5	36.0

The global matrices B_k and C_k also contain local and symmetric joint relationships. This section aims to explore the impact of other modules on the prediction results in the presence of global information. For this purpose, the effects of removing local kinematic connections $\tilde{A}_{sec}$ and symmetry $\tilde{A}_{sym}$ on the prediction of Human3.6M were studied separately, as shown in Table 2.

Table 2

Comparison of the influence of local spatial semantics

Method(T=27,CPN)	MPJPE(mm)	Δ
Ours w/o Local GCNs with A_{sec}	46.3	0.8
Ours w/o Local GCNs with A_{sym}	46.5	1.0
Ours	45.5	

Experiments have shown that removing local connections and symmetry can increase errors by 0.8 millimeters and 10 millimeters, respectively. is essential for generating more accurate 3D poses and is a supplement to global spatial semantics. From this, it can be concluded that incorporating local connections and symmetric prior knowledge.

5.3. Quantitative Analysis

This section compares with other 3D pose estimation methods on the Human3.6M dataset. For fair comparison, the same CPN detection of 2D poses is used as input as other methods. Table 3 demonstrates the results of the comparison between the model with sen-

Table 3

Comparison of estimation errors of different 3D human pose estimation algorithms on Human3.6M

	Direct	Discuss	Eating	Greet	Phone	Photo	Pose	Purch
Cai [18]	44.6	47.4	45.6	48.8	50.8	59.0	47.2	43.9
Pavllo [19]	45.2	46.7	43.3	45.6	48.1	55.1	44.6	44.3
Lee [20]	40.2	49.2	47.8	52.6	50.1	75.0	52.0	43.0
Liu [21]	41.8	44.8	41.1	44.9	47.4	54.1	43.4	42.2
Wang [22]	40.2	42.5	42.6	41.1	46.7	56.7	41.4	42.3
Liu [23]	43.3	46.1	40.9	44.6	46.6	54.0	44.1	42.9
Ours (T=243 CPN)	41.8	44.7	41.8	44.5	45.9	52.1	42.9	42.3
Ours (T=243 GT)	33.4	37.3	29.7	33.7	32.5	36.4	37.8	32.8
	Sitting	SitingD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Cai [18]	57.9	61.9	49.7	46.6	51.3	37.1	39.4	48.8
Pavllo [19]	57.3	65.8	47.1	44.0	49.0	32.8	33.9	46.8
Lee [20]	55.8	73.9	54.1	55.6	58.2	43.3	43.3	52.8
Liu [21]	56.2	63.6	45.3	43.5	45.3	31.3	32.2	45.1
Wang [22]	56.2	60.4	46.3	42.2	46.2	31.7	31.0	44.5
Liu [23]	55.3	57.9	45.8	43.4	47.3	30.4	30.3	44.9
Ours (T=243 CPN)	54.2	54.2	45.5	43.0	44.4	32.4	33.2	44.5
Ours (T=243 GT)	37.1	39.4	33.7	33.2	32.7	25.7	26.7	33.5

sory field of 243 and other methods. The results show that the method in this section achieves an improvement in accuracy with evaluation metric 1. The method in this section is close to the results of [22], which not only utilizes spatial and temporal information, but also applies pose refinement and adds motion loss to conventionally reconstructed 3D poses, whereas in this section, only spatio-temporal information is modelled through a simple network and only MPIPE loss is used without any other constraints. In addition, the table also shows the prediction results using real 2D poses as inputs, which shows an accuracy improvement of approximately 11.0 mm compared to using CPN predictions as inputs

For the HumanEva-dataset, which consists of shorter videos compared to Human3.6M, the experiments were chosen to be evaluated using a smaller receptive field27. Under evaluation index 2, the prediction results of this section are compared with other methods, as shown in Table 4, due to the corruption of the motion capture data, the prediction error of this section's method is larger in the action "Walking" of S3, except for the other actions, in which better prediction results are achieved.

In order to compare the advantages and disadvantages of this section's method with TCN, the models trained using different receptive fields from [23] and this section's method are compared in terms of the number of parameters and estimation errors, as shown in Figure 5-4. As can be seen in Figure 5-4(a)), this section's method achieves smaller estimation errors for all combinations of receptive fields on Human3.6M

Table 4

Comparison results of different 3D human gesture estimation algorithms in HumanEva-l

	Walk			Jog			Box		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
Martinez[24]	19.7	17.4	46.8	26.9	18.2	18.6			
Lee [20]	18.6	19.9	30.5	25.7	16.8	17.7	42.8	48.1	53.4
Pavllo [19]	13.8	10.2	46.5	21.0	13.1	13.5	24.1	33.2	31.6
Liu [23]	16.8	12.3	48.8	26.4	15.2	22.5	26.6	34.1	34.2
Ours	14.3	9.9	47.3	22.6	13.0	13.0	23.7	39.5	29.9

under both evaluation metrics. In addition, the model using 27 receptive fields in this section is also slightly better than the TCN using 243 receptive fields, suggesting that the use of spatial information helps to reconstruct a more accurate 3D pose. For the bars in Figure 5-4b), the model in this section uses 50.5%, 26.2% and 44.2% fewer parameters compared to the TCN's models with 9, 81 and 243 receptive fields, respectively, which suggests that the interleaved combination of the spatial and temporal mechanisms in this section's approach makes the network used for video pose estimation more efficient.

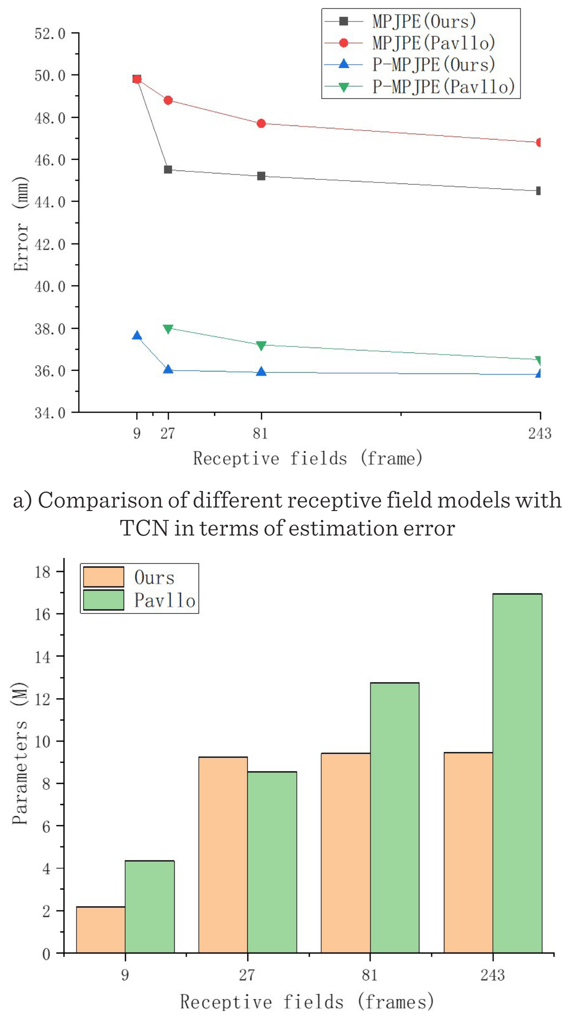
5.4. Qualitative Analysis

Based on yoga and baseball sports videos, a 3D HPE method based on visual perceptual domain is investigated. In the yoga video shown in Figure 6(a), it achieves pose reconstruction with/without local motion linkage seconds. And in the baseball video shown in Figure 6(b), the method also achieves pose reconstruction with/without the global adaptive array B and labels the pose errors as blue circles. Interestingly, in the qualitative study of the yoga video, it was found that the position estimation of the end joints was still not accurate enough even when considering the kinematic connection. In baseball videos, on the other hand, even in the face of 2D pose errors caused by occlusion, the method is still able to suppress self-occlusion using global pose semantics and time-domain information, resulting in more accurate and smoother human pose results.

The experiment also considered the situation where the camera only captured the upper body of a person, as shown in Figure 7(a). The network generated a reasonable three-dimensional pose of the upper body. Al-

Figure 6

Comparison of model performance with TCN on Human3.6M

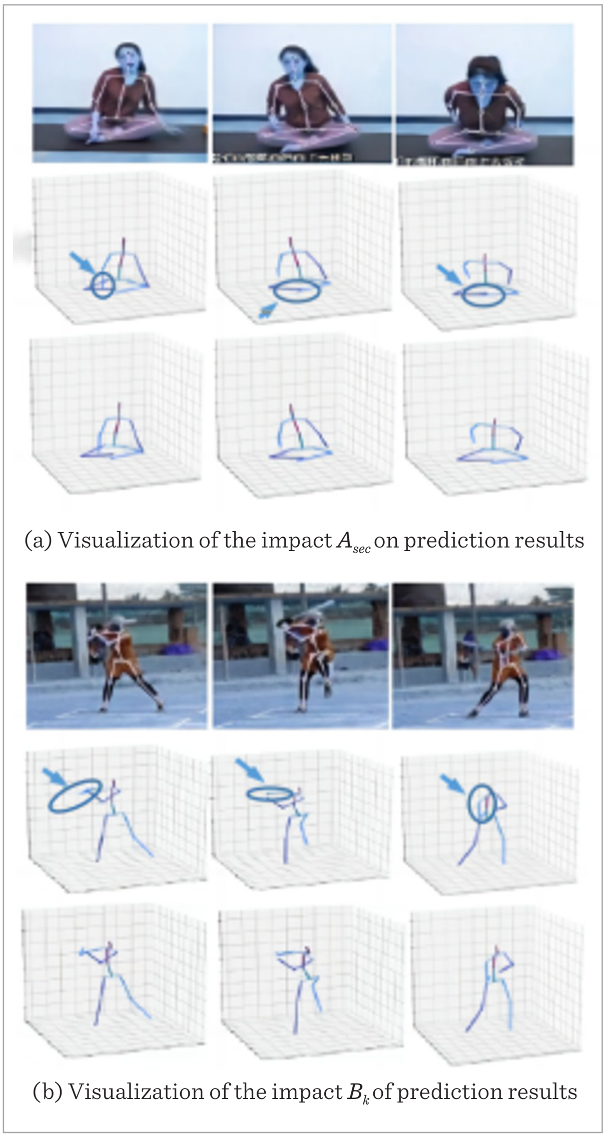


a) Comparison of different receptive field models with TCN in terms of estimation error

b) Comparison of different receptive field models with TCN in terms of number of parameters

Figure 7

Visualization of the impact of A_{sec} and B_k on the predicted outcomes

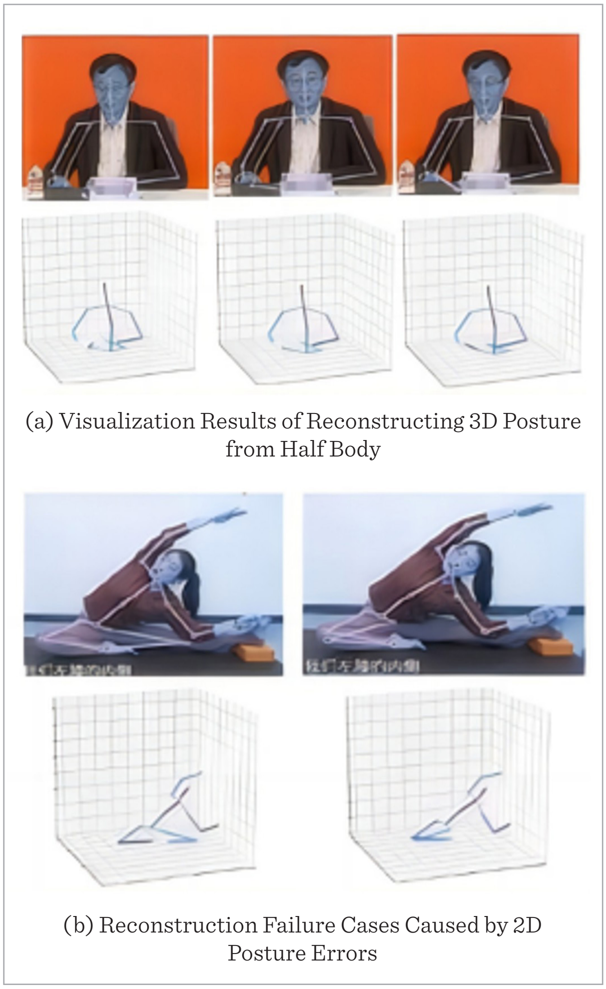


(a) Visualization of the impact A_{sec} on prediction results

(b) Visualization of the impact B_k of prediction results

Figure 8

Special scene reconstruction and failed reconstruction cases



(a) Visualization Results of Reconstructing 3D Posture from Half Body

(b) Reconstruction Failure Cases Caused by 2D Posture Errors

though the network was only trained on the complete pose, it effectively reconstructed upper body test data that it had never seen before. Figure 7(b) shows reconstruction failure cases caused by larger two-dimensional pose detection errors.

Special scene reconstruction and failed reconstruction cases can be seen in Figure 8. And to test the speed of 2D to 3D video pose estimation, this section implemented the model using different inference modes and receptive field sizes, as shown in Table 3. These tests were run on the platform mentioned previously, using local execution in the experiments, without parallel optimization of the inference. Speed comparison of different inference modes can be seen in Table 5.

Table 5

Speed comparison of different inference modes

Ours	Layer-by-Layer inference			Single-frame inference		
Receptive fields	27	81	243	27	81	243
Frames per second	18123	15703	12090	3774	2812	2190

Due to the parallel processing of input frames through layer-by-layer inference, the estimation speed is faster compared to single frame inference.

6. Conclusion

This article investigates the use of spatiotemporal graph attention mechanisms to address occlusion issues in 3D human pose estimation. It leverages 2D human skeleton information, employing mature 2D pose detectors such as stacked hourglass and cascaded pyramid models to extract 2D skeleton data from monocular images. Subsequently, a neural network is utilized to learn the mapping relationship between 2D and 3D poses, and to study how spatiotemporal information can enhance estimation accuracy. The article concludes with the following findings:

1. A human action recognition algorithm based on data-driven graph convolution and attention mechanisms (AGCN-STC) is proposed. This algorithm enhances the model's ability to express different action types and focus on important features through an adaptive graph convolution structure and multi-dimensional attention mechanisms.
2. A spatiotemporal Transformer network is constructed, which compresses sequence length progressively to focus on predicting the pose of intermediate frames. By utilizing the powerful sequence modeling capabilities of the Transformer, combined with self-attention and strided convolutions to extract spatio-temporal features, it significantly improves the accuracy of 3D pose estimation.
3. A 3D human pose estimation algorithm based on a spatiotemporal graph attention network is proposed. This algorithm captures spatial local information through graph attention and extracts temporal information through time convolutions, effectively preventing the loss of spatiotemporal information and enhancing the model's robustness, especially when dealing with occlusion issues.

Future research in 3D human pose estimation will focus on improving the performance and generalization of the model, especially in dealing with occlusion problems and improving real-time performance. Researchers can explore more advanced spatio-temporal information fusion strategies, such as combining graph convolutional networks and Transformer, to extract and fuse spatio-temporal features more efficiently. In addition, data-driven approaches, such as self-supervised learning and semi-supervised learning, will help models to optimize and generalize in the face of large amounts of data. Real-time performance improvement is also a focus of future research, where faster processing speed and lower computational cost can be achieved through model lightweighting and optimization to meet the demands of real-time applications. Cross-modal learning and the development of interactive applications will also be an important direction for future research, which will help the 3D human posture estimation technology to be widely used in human-computer interaction, virtual reality and other fields. Through the exploration of these research directions, 3D human posture estimation technology is expected to make greater breakthroughs in accuracy, robustness, real-time and other aspects, so that it can be widely used in more fields.

Acknowledgement

This work was funded by the Fuzhou Institute of Technology 2023 Research Fund Project, Project No. FTKY2023012, and I am very grateful for this financial support.

References

1. Aksan, E., Cao, P., Kaufmann, M., Cao, P., Hilliges, O. Attention, Please: A Spatio-Temporal Transformer for 3D Human Motion Prediction. Computer Vision and Pattern Recognition, 2020, 2(3), 5. arXiv Preprint arXiv:2004.08692. https://doi.org/10.1109/3DV53792.2021.00066
2. Cai, Y., Ge, L., Liu, J., Cai, J., Cham, T.-J., Yuan, J., Thalmann, N. M. Exploiting Spatial-Temporal Relationships for 3D Pose Estimation Via Graph Convolutional Networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, 2272-2281. https://doi.org/10.1109/ICCV.2019.00236
3. Chen, Z., Huang, W., Liu, H., Wang, Z., Wen, Y., Wang, S. ST-TGR: Spatio-Temporal Representation Learning for Skeleton-Based Teaching Gesture Recognition. Sensors, 2024, 24(8), 2589. https://doi.org/10.3390/s24082589

4. Davoodnia, V., Ghorbani, S., Carbonneau, M. A., Messier, A., Etemad, A. UPose3D: Uncertainty-Aware 3D Human Pose Estimation with Cross-View and Temporal Cues. Computer Vision and Pattern Recognition, 2024. arXiv Preprint arXiv:2404.14634

5. Hassanin, M., Khamiss, A., Bennamoun, M., Boussaid, F., Radwan, I. Crossformer: Cross Spatio-Temporal Transformer for 3D Human Pose Estimation. Computer Vision and Pattern Recognition, 2022. arXiv Preprint arXiv:2203.13387. https://doi.org/10.2139/ssrn.4213439

6. Jiang, W. Graph-Based Deep Learning for Communication Networks: A Survey. Computer Communications, 2022, 185, 40-54. https://doi.org/10.1016/j.comcom.2021.12.015

7. Jiang, W., Luo, J. Graph Neural Network for Traffic Forecasting: A Survey. Expert Systems with Applications, 2022, 207, 117921. https://doi.org/10.1016/j.eswa.2022.117921

8. Jiang, Z., Zhou, Z., Li, L., Chai, W., Yang, C.-Y., Hwang, J.-N. Back to Optimization: Diffusion-Based Zero-Shot 3D Human Pose Estimation. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, 6142-6152. https://doi.org/10.1109/WACV57701.2024.00603

9. Lee, K., Lee, I., Lee, S. Propagating LSTM: 3D Pose Estimation Based on Joint Interdependency. In European Conference on Computer Vision, Munich, Germany, 2018, 123-141. https://doi.org/10.1007/978-3-030-01234-2_8

10. Li, K., Wang, S., Zhang, X., Xu, Y., Xu, W., Tu, Z. Pose Recognition with Cascade Transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, 1944-1953. https://doi.org/10.1109/CVPR46437.2021.00198

11. Li, S., Chan, A. B. 3D Human Pose Estimation from Monocular Images with Deep Convolutional Neural Network. In Computer Vision--ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part II. Springer International Publishing, 2015 332-347. https://doi.org/10.1007/978-3-319-16808-1_23

12. Liu, J., Rojas, J., Liang, Z., Li, Y., Guan, Y. A Graph Attention Spatio-Temporal Convolutional Network for 3D Human Pose Estimation in Video. In IEEE International Conference on Robotics and Automation, Xi'an, China, 2021, 3374-3380. https://doi.org/10.1109/ICRA48506.2021.9561605

13. Liu, R., Shen, J., Wang, H., Chen, C., Cheung, S.-C., Asari, V. Attention Mechanism Exploits Temporal Contexts: Real-Time 3D Human Pose Reconstruction. In IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 2020, 5064-5073. https://doi.org/10.1109/CVPR42600.2020.00511

14. Liu, Y., Qiu, C., Zhang, Z. Deep Learning for 3D Human Pose Estimation and Mesh Recovery: A Survey. arXiv Preprint arXiv:2402.18844, 2024. https://doi.org/10.1016/j.neucom.2024.128049

15. Martinez, J., Hossain, R., Romero, J., Little, J. J. A Simple Yet Effective Baseline for 3D Human Pose Estimation. In IEEE International Conference on Computer Vision, Seoul, Korea, 2017, 2640-2649. https://doi.org/10.1109/ICCV.2017.288

16. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M. 3D Human Pose Estimation in Video with Temporal Convolutions and Semi-Supervised Training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, 7753-7762. https://doi.org/10.1109/CVPR.2019.00794

17. Tang, Z., Qiu, Z., Hao, Y., Hong, R., Yao, T. 3D Human Pose Estimation with Spatio-Temporal Criss-Cross Attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, 4790-4799. https://doi.org/10.1109/CVPR52729.2023.00464

18. Wang, J., Yan, S., Xiong, Y., Lin, D. Motion Guided 3D Pose Estimation from Videos. In European Conference on Computer Vision. Springer, Cham, 2020, 764-780. https://doi.org/10.1007/978-3-030-58601-0_45

19. Zhang, H., Hu, Z., Sun, Z., Zhao, M., Bi, S., Di, J. A Fused Convolutional Spatio-Temporal Progressive Approach for 3D Human Pose Estimation. The Visual Computer, 2023, 1-13. https://doi.org/10.1007/s00371-023-03088-2

20. Zhang, L., Shao, X., Li, Z., Zhou, X.-D., Shi, Y. Spatio-Temporal Attention Graph for Monocular 3D Human Pose Estimation. In 2022 IEEE International Conference on Image Processing (ICIP). IEEE, 2022, 1231-1235. https://doi.org/10.1109/ICIP46576.2022.9898019

21. Zheng, Q., Saponara, S., Tian, X., eYu, Z., Elhanashi, A., Yu, R. A Real-Time Constellation Image Classification Method of Wireless Communication Signals Based on the Lightweight Network MobileViT. Cognitive Neurodynamics, 2024, 18(2), 659-671. https://doi.org/10.1007/s11571-023-10015-7

22. Zheng, Q., Tian, X., Yu, Z., Ding, Y., Elhanashi, A., Saponara, S., Kpalma, K. MobileRaT: A Lightweight Radio

Transformer Method for Automatic Modulation Classification in Drone Communication Systems. Drones, 2023, 7(10), 596. https://doi.org/10.3390/drones7100596

23. Zheng, Q., Zhao, P., Zhang, D., Wang, H. MR-DCAE: Manifold Regularization-Based Deep Convolutional Autoencoder for Unauthorized Broadcasting Identification. International Journal of Intelligent Systems, 2021, 36(12), 7204-7238. https://doi.org/10.1002/int.22586

24. Zhou, K., Han, X., Jiang, N., Jia, K., Lu, J. Hemlets Pose: Learning Part-Centric Heatmap Triplets for Accurate 3D Human Pose Estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, 2344-2353. https://doi.org/10.1109/ICCV.2019.00243