PoseGTAC: Graph Transformer Encoder-Decoder with Atrous Convolution for 3D Human Pose Estimation

Yiran Zhu, **Xing Xu***, **Fumin Shen**, **Yanli Ji**, **Lianli Gao** and **Heng Tao Shen**
Center for Future Media & School of Computer Science and Engineering
University of Electronic Science and Technology of China, China
{yiranupup, fumin.shen}@gmail.com, {xing.xu, yanliji, lianli.gao}@uestc.edu.cn,
shenhengtao@hotmail.com

Abstract

Graph neural networks (GNNs) have been widely used in the 3D human pose estimation task, since the pose representation of a human body can be naturally modeled by the graph structure. Generally, most of the existing GNN-based models utilize the restricted receptive fields of filters and single-scale information, while neglecting the valuable multi-scale contextual information. To tackle this issue, we propose a novel model named *Graph Transformer Encoder-Decoder with Atrous Convolution* (PoseGTAC), to effectively extract multi-scale context and long-range information. Specifically, our PoseGTAC model has two key components: Graph Atrous Convolution (GAC) and Graph Transformer Layer (GTL), which are respectively for the extraction of local multi-scale and global long-range information. They are combined and stacked in an encoder-decoder structure, where graph pooling and unpooling are adopted for the interaction of multi-scale information from local to global aspect (*e.g.,* part-scale and body-scale). Extensive experiments on the Human3.6M and MPI-INF-3DHP datasets demonstrate that the proposed PoseGTAC model achieves state-of-the-art performance.

1 Introduction

In recent years, 3D human pose estimation is attracting intensive attention in various human-related research fields, such as action recognition [Yan *et al.*, 2018; Ji *et al.*, 2019], human-object interaction [Li *et al.*, 2020] and motion prediction [Mao *et al.*, 2019]. Its goal is to estimate 3D coordinates of human body joints from 2D poses or images. Compared with the methods [Zhou *et al.*, 2017; Wu and Xiao, 2020] using RGB images, the 2D-to-3D methods [Zhao *et al.*, 2019; Liu *et al.*, 2020] using only 2D poses can avoid the influence of background noise and greatly reduce the computational complexity, thus achieving competitive performance.

In this paper, we focus on the topic of *2D-to-3D pose estimation*, which aims to predict 3D poses only given the 2D pose data. Recently, as an extension of the standard convolutional network, graph neural network (GNN) has shown its natural superiority in capturing the irregular structures in visual data that CNN cannot handle. For human pose estimation, both the 2D and 3D pose information can be regarded as a graph and intuitively be modeled by GNN. The GNN-based methods take the body joints as the nodes and the bones physically connecting body joints as the edges to build the graph. Compared with the traditional methods, the GNN-based methods have achieved better performance. For example, ST-GCN [Yan *et al.*, 2018] first utilized graph convolutional network (GCN) to aggregate the skeleton features and achieved impressive performance. Later, SemGCN [Zhao *et al.*, 2019] introduced the semantic graph convolutional network to capture local and non-local information. Besides, SD-HNN [Liu *et al.*, 2020] leveraged hypergraphs to model the dynamics of the human body for 3D pose estimation.

Though the above approaches have achieved good performance in 3D human pose estimation, they generally adopt the restricted receptive fields of filters and aggregate the single-scale joint information. As illustrated in Fig. 1, existing approaches limitedly consider the 1-hop neighbors when calculating the graph convolution, while neglecting the valuable multi-scale contextual information. Actually, the multi-scale contextual information contains rich features that are essential to facilitate the prediction performance,

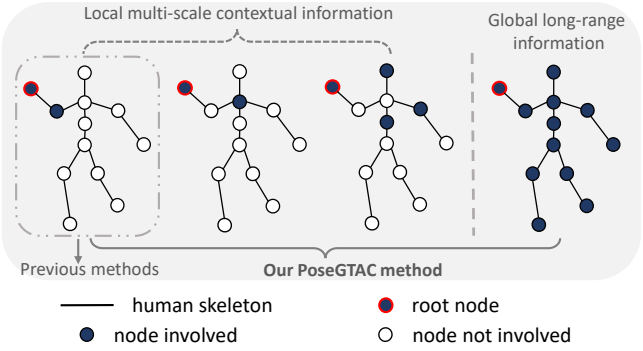


Figure 1: Previous methods commonly adopt the restricted receptive fields of filters and ignore various multi-scale contextual information. Differently, our proposed PoseGTAC method enlarges the receptive fields of filters for capturing multi-scale context.

*Corresponding author

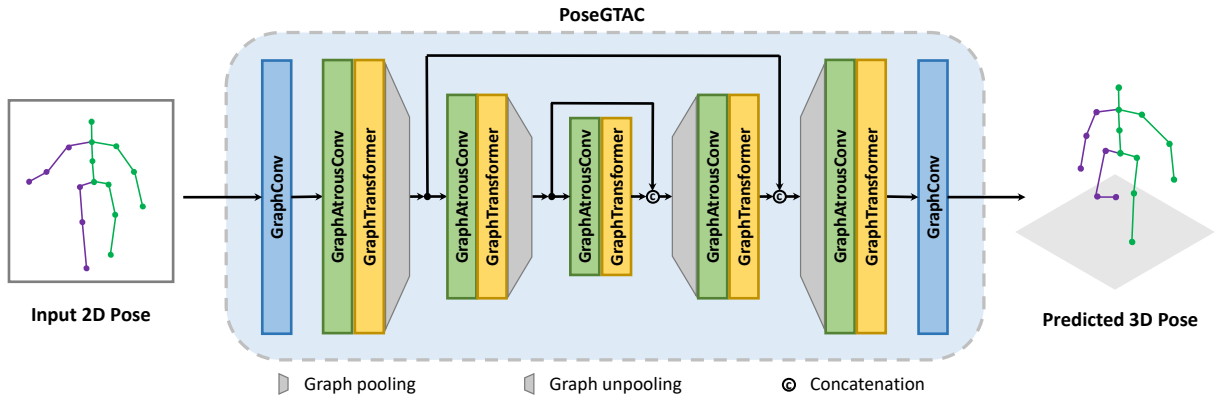


Figure 2: The overall framework of the proposed PoseGTAC model. It is a hierarchical encoder-decoder architecture that consists of stacked graph atrous convolution (GAC) layers and graph transformer layers (GTL) at different scales. In addition, at the beginning and end of the model, two graph convolution layers are used for the input encoding and output decoding procedures.

though it implicitly resides among the higher-order neighbors. Moreover, several recent approaches [Zhao *et al.*, 2019; Liu *et al.*, 2020] attempt to extract non-local information, but ignore the semantic positional information (*i.e.,* the joint type). For instance, two joints of the same coordinates with different semantics may convey totally different information.

To overcome the limitations of existing GNN-based approaches, we propose a novel *Graph Transformer Encoder-Decoder with Atrous Convolution*, dubbed *PoseGTAC*, to enhance the extraction of multi-scale context and long-range relationships in the pose. As the overall framework is shown in Fig. 2, our proposed PoseGTAC consists of stacked graph atrous convolution layers and graph transformer layers in an encoder-decoder structure which exploits multi-scale features based on human kinetics. Notably, the Graph Atrous Convolution (GAC) can effectively enlarge the receptive fields of filters and densely learn multi-scale pose context, and Graph Transformer Layer (GTL) is used to capture global long-range information. Moreover, graph pooling and graph unpooling are adopted in PoseGTAC to ensure the interaction of multi-scale information from local to global.

Our main contributions are three-fold: (1) We propose a novel PoseGTAC method that can effectively extract local multi-scale context and global long-range relationships for 3D human pose estimation. (2) We design an advanced Graph Atrous Convolution (GAC) to enlarge the receptive fields of filters and learn multi-scale pose context, and a Graph Transformer Layer (GTL) to capture global long-range relationships. The two key components are flexibly combined in an encoder-decoder structure. (3) We conduct extensive experiments on two widely-used datasets Human3.6M and MPI-INF-3DHP to demonstrate the superiority of our proposed PoseGTAC model comparing to the state-of-the-art methods.

2 Related Work

2D-to-3D Pose Estimation. With a lot of research into 3D human pose estimation, the existing methods can be grouped into three directions, 2D-to-3D pose estimation [Zhao *et al.*, 2019; Liu *et al.*, 2020], monocular image-based 3D pose estimation [Zhao *et al.*, 2019; Wu and Xiao, 2020] and multi-view image-based 3D pose estimation [Qiu *et al.*, 2019]. The recent GNN-based methods have greatly improved prediction performance in 2D-to-3D pose estimation. ST-GCN [Yan *et al.*, 2018] first applied graph convolution to aggregate features in the skeleton. SemGCN [Zhao *et al.*, 2019] utilized semantic graph convolution to lift 2D pose to 3D pose by extracting local and non-local information. SD-HNN [Liu *et al.*, 2020] introduced static and dynamic hypergraphs to represent a human body for 3D pose estimation. However, they do not take full advantage of multi-scale contextual information.

Graph Neural Networks. As a generalization of standard convolution and pooling, many methods of graph convolution and graph pooling have recently been proposed. Inspired by the graph Laplacian methods, [Kipf and Welling, 2017] proposed graph convolution networks by the Chebyshev approximation, which is the most widely used form of graph convolution. GraphSAGE [Hamilton *et al.*, 2017] embedded node features by sampling and aggregating and introduced transductive graph convolution. SAGPool [Lee *et al.*, 2019] attempted to use a learnable mask to select the node features retained. In this work, we propose a novel graph transformer encoder-decoder with atrous convolution.

Attention Mechanism. The pioneering work of [Vaswani *et al.*, 2017] introduced a new attention based network named Transformer for the machine translation task of natural language processing (NLP). Transformer is mainly composed of multi-head attention module and feedforward network. There has also been a lot of work based on attention mechanism in computer vision recently [Wang *et al.*, 2018; Carion *et al.*, 2020]. Unlike non-local network based methods [Zhao *et al.*, 2019], our PoseGTAC method introduces the transformer operation on the graph to capture long-range relationships.

3 Proposed Method

3.1 Preliminaries

Graph Definition. The raw pose data, *i.e.,* the joint keypoint vector, is a set of 2D coordinates. Usually, in GNN-based approaches, the vector can be represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertices are all joints and edges are physical

connections between two joints. Here $\mathcal{V}$ is the set of N joints and $\mathcal{E}$ is characterized by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$. In this way, the pose data is transformed into a graph sequence and specifically represented as a tensor $\mathbf{X} \in \mathbb{R}^{N\times C}$, where N and C denote the numbers of joints and channels.

Graph Convolution. Based on the above definition, the existing GNN-based methods generally use the stacked graph convolution module to extract the high-level skeleton information to regress 3D pose. The commonly used graph convolution operation can be represented as follows:

$$\mathbf{X}^{(i+1)} = \sigma(\mathbf{W}\mathbf{X}^{(i)}(\mathbf{\Lambda} \odot \mathbf{M})), \tag{1}$$

where $\sigma(\cdot)$ denotes the activation function, i is the index of the current layer and $\mathbf{\Lambda}$ denotes the normalized Laplace matrix. $\mathbf{\Lambda} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{A}$ is the adjacency matrix with the self-loop, and $\mathbf{D}$ denotes the degree matrix of the graph. $\mathbf{W}$ denotes a $C_{out} \times C_{in} \times 1 \times 1$ learnable weight matrix and $\mathbf{M}$ denotes an $N \times N$ attention mask matrix. $\odot$ denotes the element-level dot product.

In addition to using local GCN to extract skeleton features, non-local module is introduced to extract long-range information, which is generally represented as extracting the relationships between the current node and all other nodes.

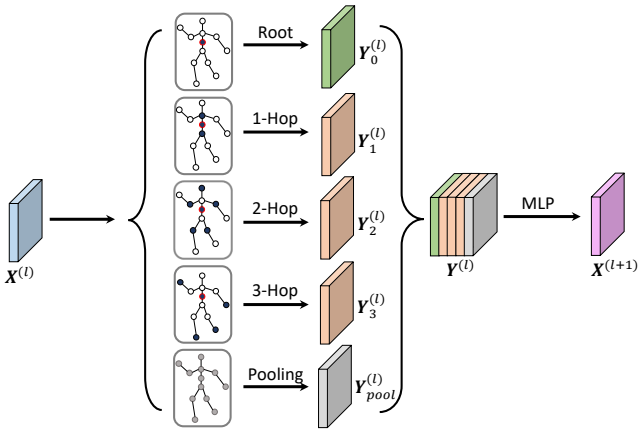


Figure 3: Illustration of the graph atrous convolution (GAC) layer, which consists of paralleled graph convolutions and graph pooling. Here "MLP" denotes multi-layer perceptron.

3.2 Graph Atrous Convolution (GAC)

As shown in Eq. 1, the previous methods use the restricted filters to convolve only 1-hop neighbors, ignoring the multi-scale contextual information from higher-order neighbors. We consider that multi-scale context is indispensable for 3D human pose estimation. To this end, we introduce a multi-scale graph convolution termed Graph Atrous Convolution (GAC) to capture the multi-scale context residing in the higher-order neighbors. Inspired by the atrous convolution [Yu and Koltun, 2016; Chen *et al.*, 2018] in image segmentation, convolution operations with different dilation factors are used in parallel. As illustrated in Fig. 3, in our graph convolution, the dilation factor is defined as the distance to the root node, and graph atrous convolution is represented as paralleled convolutions with root node, 1-hop, 2-hop and 3-hop neighbors, etc. We first formally define the k-hop matrix $\mathbf{A}_k$ as follows:

$$[\mathbf{A}_k]_{i,j} = \begin{cases} 1 & d(v_i, v_j) = k, \\ 1 & d(v_i, v_j) = 0, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where $d(v_i, v_j)$ denotes the distance of the shortest path between v_i and v_j on the skeleton graph, and $\mathbf{A}_k$ is the k-hop adjacency matrix with the self-loop.

$$\mathbf{Y}_k^{(l)} = \sigma(\mathbf{W}_k\mathbf{X}^{(l)}(\mathbf{\Lambda}_k \odot \mathbf{M}_k)), \tag{3}$$

where $\mathbf{\Lambda}_k$ denotes the normalized k-hop Laplace matrix. $\mathbf{\Lambda}_k = \mathbf{D}_k^{-\frac{1}{2}}\mathbf{A}_k\mathbf{D}_k^{-\frac{1}{2}}$, where $\mathbf{D}_k$ denotes the degree matrix of the graph. $\mathbf{W}_k$ denotes a learnable weight matrix for node embedding and $\mathbf{M}_k$ denotes a $N\times N$ learnable attention mask matrix. $\mathbf{Y}_k^{(l)} \in \mathbb{R}^{N\times C}$ denotes the output of k-hop graph atrous convolution.

Moreover, in order to facilitate global context information, skeleton features pooled globally are concatenated with the output of parallel graph atrous convolution in Eq. 3 and then fed to a multi-layer perceptron (MLP) to aggregate multi-scale and global context features.

$$\begin{cases} \mathbf{Y}_{pool}^{(l)} = \text{AvgPool}(\mathbf{X}^{(l)}), \\ \mathbf{Y}^{(l)} = \text{Cat}([\mathbf{Y}_0^{(l)}, ..., \mathbf{Y}_{k-1}^{(l)}, \mathbf{Y}_{pool}^{(l)}]), \\ \mathbf{X}^{(l+1)} = \mathbf{W}\mathbf{Y}^{(l)}, \end{cases} \tag{4}$$

where AvgPool$(\cdot)$ and Cat$(\cdot)$ respectively denote the average pooling and concatenation operation. $\mathbf{Y}_{pool}^{(l)} \in \mathbb{R}^{N\times C}$ represents the output features pooled globally, $\mathbf{Y}^{(l)} \in \mathbb{R}^{N\times[(k+1)\times C]}$ is the features of concatenating all branches. $\mathbf{W}$ is a learnable weight matrix for feature aggregation and dimension reduction. $\mathbf{X}^{(l+1)}$ denotes the final output features of the GAC layer.

3.3 Graph Transformer Layer (GTL)

Although the multi-scale contextual information has been well extracted by the GAC module based on the local physical connection of the human body, there is a lack of long-range information that can effectively promote pose representation learning. To determine whether there is a connection between two joints and how strong the connection is, we introduce the graph transformer layer (GTL) to better capture long-range information.

Since the joints in the pose have no order to uniquely identify their types from the input, the position encoding needs to be added to complement the position information. In particular, we follow the sine and cosine functions as the position encoding functions in [Vaswani *et al.*, 2017]:

$$\begin{aligned} PE_{(pos,2i)} &= \sin(pos/10000^{2i/C_{in}}), \\ PE_{(pos,2i+1)} &= \cos(pos/10000^{2i/C_{in}}), \end{aligned} \tag{5}$$

where pos is the position and i is the dimension of the position encoding vector. As shown in Fig. 4, in the GTL, the raw

input is first added with the position encoding, and then fed to two embedding functions (*e.g.,* θ and ϕ) for obtaining the high-level features. The dot product is adopted to measure the similarity of the two joints in an embedding space. Specifically, we can calculate the attention matrix representing the strength of the relationships between the joints as follows:

$$\mathbf{M_{att}} = \text{Softmax}(\mathbf{X_{in}}^T \mathbf{W}_\theta^T \mathbf{W}_\phi \mathbf{X_{in}}), \tag{6}$$

where $\text{Softmax}(\cdot)$ denotes the softmax operation used for normalization, and $\mathbf{M_{att}}$ denotes the attention map. $\mathbf{W}_\theta$ and $\mathbf{W}_\phi$ are the learnable weight matrices of the embedding functions θ and ϕ, respectively.

In addition, we add an extra $N \times N$ global attention matrix $\mathbf{M_{global}}$ to pay more attention on unconstrained learning. Specifically, the global attention matrix is added to the attention matrix $\mathbf{M_{att}}$ and then multiplied by the original input. As a result, the multi-head attention and feedforward network can be used to obtain long-range features with rich attention.

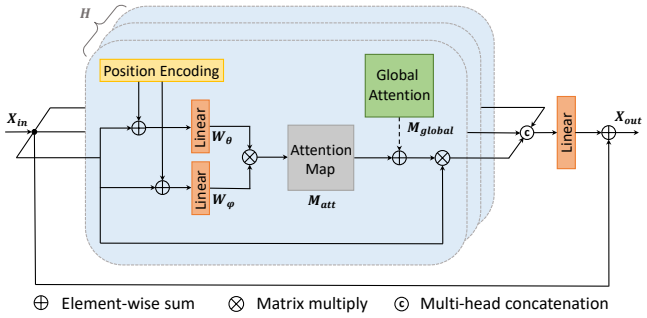


Figure 4: Illustration of the graph transformer layer (GTL) that is composed of H self-attention modules and a feedforward network.

3.4 The PoseGTAC Architecture

To obtain multi-scale (*e.g.,* joint-scale and part-scale) information based on human kinetics, graph pooling and graph unpooling are adopted to capture effectively the interaction of multi-scale information in the pose. Before graph pooling and unpooling, the nodes of each scale s are divided into different regions $\mathcal{R}^s$ according to the physical priors of the human body, such as the upper left leg, the lower left leg, the torso, etc. For the upper scale features, we implement the pooling by using average pooling to aggregate the point features that are divided into a region in the current scale to get the lower scale features. In other words, multiple points are averaged into one point. Specifically, the formula is implemented as follows:

$$\mathbf{X}_i^{s+1} = \text{AvgPool}(\{\mathbf{X}_j^s \mid \forall\, \mathbf{X}_j^s \in \mathcal{R}_i^s\}), \tag{7}$$

where $\mathcal{R}_i^s$ denotes the region feature under scale s, and $\mathbf{X}_j^s$ denotes a joint feature element belonging to $\mathcal{R}_i^s$ set. $\mathbf{X}_i^{s+1}$ denotes a new joint feature under scale $s+1$ obtained through graph pooling.

Due to the independence of the partitioned regions, the unpooling is realized by copying an upper scale feature multiple times and then concatenating them together in the corresponding lower scale region.

$$\mathcal{R}_i^s = \text{Cat}(\{\mathbf{X}_i^{s+1}, \ldots, \mathbf{X}_i^{s+1}\}), \tag{8}$$

where the number of repetitions of $\mathbf{X}_i^{s+1}$ is determined by the size of the corresponding set $\mathcal{R}_i^s$.

In addition, the features obtained by unpooling are concatenated with the corresponding features on the contraction path, and then fed to the next layer. The graph atrous convolution layer and graph transformer layer are used to extract local and global information, while pooling and unpooling are used to facilitate the interaction of information from local to global. Fig. 2 illustrates the architecture of our proposed PoseGTAC, which stacks five graph atrous convolution layers and five graph transformer layers at different scales. Two graph convolution layers are used for input encoding and output decoding procedures. Each layer is followed by a BN layer and a ReLU layer. The mean squared error (MSE) between the predicted pose and the ground-truth is used as our loss function, which can be trained in an end-to-end manner.

4 Experiments

4.1 Experimental Setup

Datasets and Evaluation Protocols. Following previous studies [Liu *et al.*, 2020], we adopt two benchmark datasets Human3.6M and MPI-INF-3DHP in our experiments.

Human3.6M [Ionescu *et al.*, 2014] is the largest 3D human pose estimation dataset. It contains 3.6 million images, where 11 professional actors perform 15 actions such as walking, greeting, smoking and making a phone call. Both 2D and 3D ground-truth data are available for supervised 3D human pose estimation. We use five subjects (S1, S5, S6, S7, and S8) for training and two subjects (S9 and S11) for testing. In order to reduce redundancy, we follow [Zhao *et al.*, 2019] and downsample the raw videos from 50fps to 10fps for the training and testing datasets. *MPI-INF-3DHP* [Mehta *et al.*, 2017] is the dataset obtained using MoCap system for 3D human pose estimation. The test set consists of 2,929 frames from 6 subjects performing 7 actions.

Three evaluation protocols are adopted in our experiments: *Protocol #1* is the mean per-joint position error (MPJPE) in millimeters which measures the error between the ground-truth and predictions. *Protocol #2* is P-MPJPE which reports the error between the ground-truth and predictions through the rigid transformation including translation, rotation and scale. As an auxiliary, we also measure the mean per-joint velocity error (MPJVE) as *Protocol #3*, which is obtained by the MPJPE of the first derivative and represents the smoothness of predicted results. Two standard metrics of PCK (Percentage of Correct Keypoints) under 150mm radius and AUC (Area under the ROC Curve) are used for quantitative evaluation for MPI-INF-3DHP.

Implement Details. Our PoseGTAC model consists of two graph convolution layers, five graph transformer layers (GTL) with three heads and five graph atrous convolution (GAC) layers. The number of channels for all layers is 128, except for GTL, which has 32 intermediate channels for reducing the computational complexity of the model. We set up three different scales containing 16,10 and 5 joints. The Adam optimizer is adopted as our optimizer with the initial learning rate 0.001 and the decay factor 0.96 per 100K steps. We train our model for 50 epochs with the batch size 256. All exper-

Method	Direct	Discuss	Eating	Greet	Phone	Photo	Pose	Purch	Sitting	SittingD	Smoke	Wait	WalkD	Walk	WalkT	Avg. ↓
Pavlakos [Pavlakos *et al.*, 2017]	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Fang [Fang *et al.*, 2018]	50.1	54.3	57.0	57.1	66.6	73.3	53.4	55.7	72.8	88.6	60.3	57.7	62.7	47.5	50.6	60.4
Yang [Yang *et al.*, 2018]	51.5	58.9	50.4	57.0	62.1	65.4	49.8	52.7	69.2	85.2	57.4	58.4	43.6	60.1	47.7	58.6
Lee [Lee *et al.*, 2018]	43.8	51.7	48.8	53.1	52.2	74.9	52.7	44.6	56.9	74.3	56.7	66.4	47.5	68.4	45.6	55.8
Trumble [Trumble *et al.*, 2018]	41.7	43.2	52.9	70.0	64.9	83.0	57.3	63.5	61.0	95.0	70.0	62.3	66.2	53.7	52.4	62.5
Chen [Chen *et al.*, 2019]	45.9	53.5	50.1	53.2	61.5	72.8	50.7	49.4	68.4	82.1	58.6	53.9	57.6	41.1	46.0	56.9
Wandt [Wandt and Rosenhahn, 2019]	50.0	53.5	44.7	51.6	49.0	58.7	48.8	51.3	51.1	66.0	46.6	50.6	42.5	38.8	60.4	50.9
Zhao [Zhao *et al.*, 2019]	37.8	49.4	37.6	40.9	45.1	41.4	40.1	48.3	50.1	**42.2**	53.5	44.3	40.5	47.3	39.0	43.8
Zhou [Zhou *et al.*, 2019]	**34.4**	42.4	36.6	42.1	38.2	**39.8**	**34.7**	40.2	45.6	60.8	39.0	42.6	42.0	**29.8**	**31.7**	39.9
Wu [Wu and Xiao, 2020]	34.9	**40.8**	37.5	47.2	41.5	46.6	35.9	39.5	52.6	72.5	42.3	45.8	42.0	31.6	33.8	43.2
Liu [Liu *et al.*, 2020]	42.1	45.6	38.2	41.4	41.5	47.4	45.8	39.9	44.7	53.0	42.6	44.0	42.1	34.0	37.6	42.7
PoseGTAC (Ours)	37.2	42.2	**32.6**	**38.6**	**38.0**	44.0	40.7	**35.2**	**41.0**	45.5	**38.2**	**39.5**	**38.2**	**29.8**	33.0	**38.2**

Table 1: Quantitative evaluation using Mean Per Joint Position Error (MPJPE) in millimeter between estimated pose and the ground-truth on Human3.6M under *Protocol #1*. The best results are in bold and the second-best results are underlined.

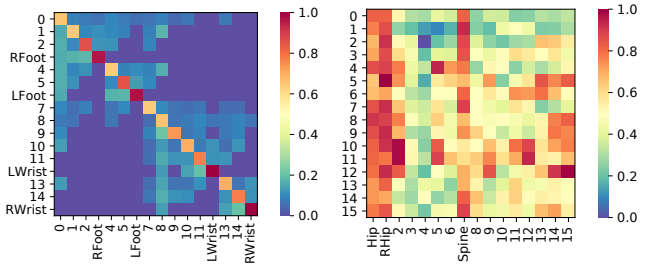
Method	Direct	Discuss	Eating	Greet	Phone	Photo	Pose	Purch	Sitting	SittingD	Smoke	Wait	WalkD	Walk	WalkT	Avg. ↓
Lee [Lee *et al.*, 2018]	38.0	39.3	46.3	44.4	49.0	55.1	40.2	41.1	53.2	68.9	51.0	39.1	33.9	56.4	38.5	46.2
Fang [Fang *et al.*, 2018]	38.2	41.7	43.7	44.9	48.5	55.3	40.2	38.2	54.5	64.4	47.2	44.3	47.3	36.7	41.7	45.7
Rayat [Hossain and Little, 2018]	35.2	40.8	37.2	37.4	43.2	44.0	38.9	35.6	42.3	44.6	39.7	39.7	40.2	32.8	35.5	39.2
Chen [Chen *et al.*, 2019]	36.5	41.0	40.9	43.9	45.6	53.8	38.5	37.3	53.0	65.2	44.6	40.9	44.3	32.0	38.4	44.1
Wandt [Wandt and Rosenhahn, 2019]	33.6	38.8	32.6	37.5	36.0	44.1	37.8	34.9	39.2	52.0	37.5	39.8	34.1	40.3	34.9	38.2
Zhou [Zhou *et al.*, 2019]	29.1	34.9	29.9	32.6	31.2	**32.3**	**27.0**	33.3	37.6	45.9	32.2	31.5	34.5	**22.9**	**25.9**	32.1
Wu [Wu and Xiao, 2020]	29.9	33.6	31.4	37.1	33.9	36.8	28.4	30.7	42.6	52.2	35.3	35.2	34.0	24.9	27.9	34.6
Liu [Liu *et al.*, 2020]	29.6	34.9	31.7	31.6	32.9	37.4	33.3	30.5	37.6	43.0	34.2	34.3	33.2	27.0	29.2	33.4
PoseGTAC (Ours)	**25.8**	**31.7**	**25.8**	**29.3**	**28.8**	34.1	29.6	**26.4**	**33.2**	**37.2**	**30.5**	**30.0**	**29.8**	23.4	**25.9**	**29.4**

Table 2: Quantitative evaluation using P-MPJPE in millimeter between estimated pose and the ground-truth on Human3.6M under *Protocol #2*. Procrustes alignment is used to preprocess the ground-truth. The best results are in bold and the second-best results are underlined.

iments are conducted on PyTorch deep learning framework with a single RTX-2080Ti GPU.

Method	PCK ↑	AUC ↑
Zhou [Zhou *et al.*, 2017]	69.2	32.5
Yang [Yang *et al.*, 2018]	69.0	32.0
Pavlakos [Pavlakos *et al.*, 2018]	71.9	35.3
Habibie [Habibie *et al.*, 2019]	70.4	36.0
Liu [Liu *et al.*, 2020]	74.9	37.5
PoseGTAC (Ours)	**76.4**	**39.3**

Table 3: Quantitative evaluation on MPI-INF-3DHP dataset using PCK and AUC. The higher values mean better performance. The best results are in bold and the second-best results are underlined.



(a) Adjacency matrix of GAC. (b) Attention matrix of GTL.

Figure 5: The learned weight matrices of our PoseGTAC model: (a) the adjacency matrix of graph atrous convolution (GAC) and (b) the attention matrix of graph transformer layer (GTL).

4.2 Quantitative Results

To better evaluate the performance of our proposed PoseGTAC model, we show quantitative results and compare them to the state-of-the-art methods on 3D human pose estimation.

Table 1 and Table 2 show the experiment results on the Human3.6M dataset for *Protocol #1* and *Protocol #2*, respectively. We can observe that the proposed PoseGTAC method outperforms all the compared models on both two protocols and obtains the best results in terms of the average and most individual actions. In particular, the MPJPE reduces by $1.7mm$ with an error reduction of 4.3%, and the P-MPJPE decreases by $2.7mm$ with an 8.4% error reduction, respectively. It proves that our PoseGTAC extracts rich multi-scale context and promotes long-range feature interaction.

Moreover, we also provide supplementary to evaluate the smoothness of the predicted pose by our PoseGTAC model as *Protocol #3*. Additionally, the results on *Protocol #3* also reflect the effectiveness and precision of our model from the side. As reported in Table 4, the MPJVE in our PoseGTAC model reduces the results of the previous work [Pavllo *et al.*, 2019] by 80%, which clearly validates that the pose predicted by our PoseGTAC model is smoother and more precise.

For the MPI-INF-3DHP dataset, we train our model with Human3.6M data without post-process of fine-tuning or retraining. As shown in Table 3, comparing to the best counterpart of [Liu *et al.*, 2020], our PoseGTAC model achieves an improvement of 1.5% PCK and 1.8% AUC, which consistently indicates the effectiveness of our PoseGTAC model.

4.3 Ablation Studies

Effectiveness of Each Module. To verify the effectiveness of different modules in our proposed model, we conduct a series of ablation studies on the Human3.6M dataset under *Protocol #1*. For the ablation of the overall network architecture, we take *SemGCN* [Zhao *et al.*, 2019] as our baseline. We set whether to use or combine graph transformer layer (GTL) and graph atrous convolution (GAC), and finally com-

Method	Direct	Discuss	Eating	Greet	Phone	Photo	Pose	Purch	Sitting	SittingD	Smoke	Wait	WalkD	Walk	WalkT	Avg. ↓
Pavllo* [Pavllo *et al.*, 2019]	12.8	12.6	10.3	14.2	10.2	11.3	11.8	11.3	8.2	10.2	10.3	11.3	13.1	13.4	12.9	11.6
PoseGTAC (Ours)	**2.3**	**2.6**	**1.9**	**2.9**	**1.9**	**2.1**	**2.3**	**2.4**	**1.3**	**1.7**	**1.8**	**2.2**	**3.4**	**3.6**	**2.9**	**2.3**

Table 4: Quantitative evaluation using Mean Per Joint Velocity Error (MPJVE) between estimated pose and the ground-truth on Human3.6M under *Protocol #3*. The best results are in bold. * denotes the methods based on single frame.

Method	MPJPE ↓	Branches	MPJPE ↓
GTL w/o PE & GA	41.5	4s	39.3
GTL + PE	40.2	4s+pool	**38.9**
GTL + PE + GA	**39.4**	6s	39.6

Table 5: The MPJPE obtained by our proposed model with different configurations of GAC and GTL.

Method	MPJPE (mm) ↓
SemGCN [Zhao *et al.*, 2019]	43.8
with GAC	38.9
with GTL	39.4
with GAC & GTL	38.6
PoseGTAC (Ours)	**38.2**

Table 6: Ablation study on the Human3.6M dataset for the MPJPE between the predicted pose and the ground-truth.

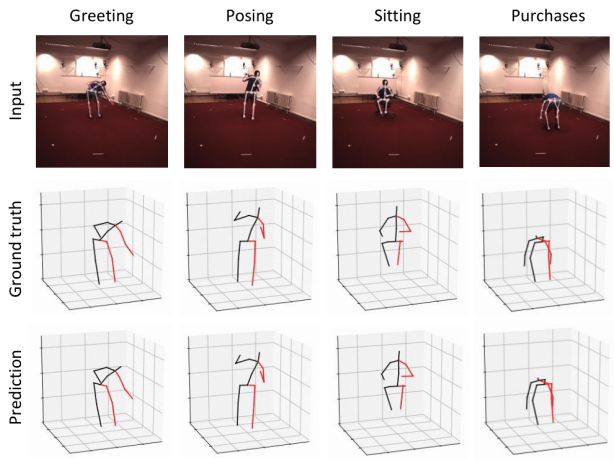


Figure 6: Qualitative results obtained by our proposed PoseGTAC on the Human3.6M dataset.

pare our PoseGTAC model with the encoder-decoder framework. As shown in Table 6, compared to our baseline, adding only GAC or GTL is able to obtain remarkable error reduction, and adopting the encoder-decoder framework achieves the best results.

We further explore the configuration of the two main modules we designed in detail. Table 5 shows the performance comparison of GAC module with different numbers of branches and whether the pooling is used in GAC. We configure our GAC with four branches (*i.e.,* "4s") and six branches (*i.e.,* "6s"). By contrast, the error of our GAC with four branches is the lowest, and then is further reduced by adding pooling (*i.e.,* "4s+pool"). We also evaluate the components of GTL, including the position encoding (PE) and global attention (GA). Based on our "vanilla" GTL, adding the position encoding or global attention can obtain 1.3% and 0.8% error reduction, showing that the position encoding can well complement the semantic information in the joint sequence.

Visualization of the Learned Matrices. Furthermore, to explore how the information is aggregated between the joints, we visualize the learned weight matrices of the first GAC and GTL of our PoseGTAC model. Specifically, we obtain the final adjacency matrix in Fig. 5(a) by adding the weight matrices of the four branches of GAC. From the adjacency matrix, *i.e.,* the GAC filter, we can see that the receptive field of our filter is sufficiently enlarged and contains all joints from the root to the 3-hop neighbors. It is probable that our model focuses more on the wrists and feet that incorporate multi-scale information. In addition, the attention matrix is obtained by adding the attention map and global attention matrix in GTL. As shown in Fig. 5(b), long-range relationships can be effectively extracted in each joint, especially in the hip and spine. It indicates that our proposed model not only focuses on local multi-scale contextual information, but also concentrate on global long-range relationships.

Qualitative Pose Estimation Results. We finally illustrate typical visualizations of the poses predicted by our PoseGTAC model in Fig. 6, where five different actions, "greeting", "posing", "sitting", and "purchases" are included. We can observe that the poses for all sequences predicted by PoseGTAC are considerably accurate comparing to the ground-truth annotations, even in the challenging scenario with occlusion problem (*e.g.,* "purchases").

5 Conclusion

In this work, we proposed a novel Graph Transformer Encoder-Decoder with Atrous Convolution named PoseGTAC to extract effectively multi-scale contextual information and capture accurately global long-range relationships for 3D human pose estimation. Moreover, we designed two modules, graph atrous convolution (GAC) and graph transformer layer (GTL), respectively for the extraction of multi-scale and long-range information, and combine them in the encoder-decoder structure. The extensive experiments on the Human3.6M and MPI-INF-3DHP datasets validated that our PoseGTAC model can extract abundant multi-scale and global long-range information, which is beneficial for 3D human pose estimation.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 61976049, 62072080 and U20B2063; the Sichuan Science and Technology Program 2018GZDZX0032, 2019ZDZX0008, 2019YFG0003, 2019YFG0533 and 2020YFS0057.

References

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020.

[Chen *et al.*, 2018] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 833–851, 2018.

[Chen *et al.*, 2019] Xipeng Chen, Kwan-Yee Lin, Wentao Liu, Chen Qian, and Liang Lin. Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation. In *CVPR*, pages 10895–10904, 2019.

[Fang *et al.*, 2018] Haoshu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning knowledge-guided pose grammar machine for 3d human pose estimation. In *AAAI*, 2018.

[Habibie *et al.*, 2019] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Gerard Pons-Moll, and Christian Theobalt. In the wild human pose estimation using explicit 2d features and intermediate 3d representations. In *CVPR*, pages 10905–10914, 2019.

[Hamilton *et al.*, 2017] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, 2017.

[Hossain and Little, 2018] Mir Rayat Imtiaz Hossain and James J. Little. Exploiting temporal information for 3d human pose estimation. In *ECCV*, pages 69–86, 2018.

[Ionescu *et al.*, 2014] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1325–1339, 2014.

[Ji *et al.*, 2019] Yanli Ji, Yue Zhan, Yang Yang, Xing Xu, Fumin Shen, and Heng Tao Shen. A knowledge map guided coarse-to-fine action recognition. In *IEEE Trans. Image Processing*, 2019.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Lee *et al.*, 2018] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating LSTM: 3d pose estimation based on joint interdependency. In *ECCV*, pages 123–141, 2018.

[Lee *et al.*, 2019] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *ICML*, 2019.

[Li *et al.*, 2020] Yonglu Li, Xinpeng Liu, Han Lu, Shiyi Wang, Junqi Liu, Jiefeng Li, and Cewu Lu. Detailed 2d-3d joint representation for human-object interaction. In *CVPR*, pages 10163–10172, 2020.

[Liu *et al.*, 2020] Shengyuan Liu, Pei Lv, Yuzhen Zhang, Jie Fu, Junjin Cheng, Wanqing Li, Bing Zhou, and Mingliang Xu. Semi-dynamic hypergraph neural network for 3d pose estimation. In *IJCAI*, pages 782–788, 2020.

[Mao *et al.*, 2019] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *ICCV*, pages 9488–9496, 2019.

[Mehta *et al.*, 2017] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved CNN supervision. In *3DV*, pages 506–516, 2017.

[Pavlakos *et al.*, 2017] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*, pages 1263–1272, 2017.

[Pavlakos *et al.*, 2018] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *CVPR*, pages 7307–7316, 2018.

[Pavllo *et al.*, 2019] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *CVPR*, pages 7753–7762, 2019.

[Qiu *et al.*, 2019] Haibo Qiu, Chunyu Wang, Jingdong Wang, Naiyan Wang, and Wenjun Zeng. Cross view fusion for 3d human pose estimation. In *ICCV*, pages 4341–4350, 2019.

[Trumble *et al.*, 2018] Matthew Trumble, Andrew Gilbert, Adrian Hilton, and John P. Collomosse. Deep autoencoder for combined human pose estimation and body model upscaling. In *ECCV*, pages 800–816, 2018.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[Wandt and Rosenhahn, 2019] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *CVPR*, pages 7782–7791, 2019.

[Wang *et al.*, 2018] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018.

[Wu and Xiao, 2020] Haiping Wu and Bin Xiao. 3d human pose estimation via explicit compositional depth maps. In *AAAI*, pages 12378–12385, 2020.

[Yan *et al.*, 2018] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.

[Yang *et al.*, 2018] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy S. J. Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *CVPR*, pages 5255–5264, 2018.

[Yu and Koltun, 2016] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[Zhao *et al.*, 2019] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *CVPR*, pages 3425–3435, 2019.

[Zhou *et al.*, 2017] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: A weakly-supervised approach. In *ICCV*, pages 398–407, 2017.

[Zhou *et al.*, 2019] Kun Zhou, Xiaoguang Han, Nianjuan Jiang, Kui Jia, and Jiangbo Lu. Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation. In *ICCV*, pages 2344–2353, 2019.