Pattern Recognition 152 (2024) 110446

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

DGFormer: Dynamic graph transformer for 3D human pose estimation

Zhangmeng Chen [a,b], Ju Dai [b,*], Junxuan Bai [c], Junjun Pan [a,b]

[a] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China
[b] Peng Cheng Laboratory, Shenzhen, China
[c] Capital University of Physical Education and Sports, Beijing, China

ARTICLE INFO

Keywords:
3D human pose estimation
Transformer
Graph

ABSTRACT

Despite the significant progress for monocular 3D human pose estimation, it still faces challenges due to self-occlusions and depth ambiguities. To tackle those issues, we propose a novel Dynamic Graph Transformer (DGFormer) to exploit local and global relationships between skeleton joints for pose estimation. Specifically, the proposed DGFormer mainly consists of three core modules: Transformer Encoder (TE), immobile Graph Convolutional Network (GCN), and dynamic GCN. TE module leverages the self-attention mechanism to learn the complex global relationships among skeleton joints. The immobile GCN is responsible for capturing the local physical connections between human joints, while the dynamic GCN concentrates on learning the sparse dynamic K-nearest neighbor interactions according to different action poses. By building the adequately global long-range, local physical, and sparse dynamic dependencies of human joints, experiments on Human3.6M and MPI-INF-3DHP datasets demonstrate that our method can predict 3D pose with lower errors outperforming the recent state-of-the-art image-based performance. Furthermore, experiments on in-the-wild videos demonstrate the impressive generalization abilities of our method. Code will be available at: https://github.com/czmmmm/DGFormer.

1. Introduction

3D human pose estimation (3D-HPE) has become an increasing research hotspot in the computer vision and graphics community because of its wide range of applications in video surveillance [1], human–robot interaction [2], motion analysis [3], etc. The goal is to estimate the 3D coordinates of human body joints. In spite of the fact that 3D-HPE has achieved considerable development [4,5] thanks to the powerful learning capabilities of deep learning, there are still challenges such as self-occlusions and depth ambiguities where multiple 3D poses correspond to the single 2D projection.

To address the above difficulties, current works can be split into two categories: (1) direct estimation and (2) 2D-to-3D lifting. Direct estimation methods [3,6] estimate 3D poses from 2D images directly via convolutional neural networks (CNN). However, the convolutional operations focus on local features, and the direct regression without intermediate process makes the computation seriously heavy. Hence, several approaches [7,8] attempt to adopt the 2D-to-3D lifting manner, which detects 2D keypoints using 2D pose detector [9] firstly and then lifts them to 3D with the popular Fully Connected Network (FCN). The FCN-based works capture the global relationships by flattening all joints while neglecting the spatial structure of the human skeleton. In fact, both the CNN-based and FCN-based methods hardly model complex poses with limited representational capacities of graph-structured data.

Graph Convolutional Network (GCN) is a promising approach to process graph-structured data. The human skeleton can be regarded as a graph with physically connected joints. To incorporate the prior, some researches [10,11] utilize GCN to explore local physical relationships for pose estimation. SemGCN [10] further introduces an extra non-local module to capture global relationships. However, existing GCN-based methods are restricted by the receptive field of the fixed physical affinity with natural connections. For diverse human actions, dynamic long-range interactions between joints also provide informative clues. For instance, as shown in Fig. 1, the joints of two arms of the *Greeting* behavior are closely related, but there is no natural connection between them. Therefore, it will facilitate performance to simultaneously consider the local physical connections and sparse long-range contextual information between joints.

Recently, the transformer [12] has dominated natural language processing (NLP) because of its powerful long-range modeling abilities. By leveraging the self-attention mechanism, the transformer model is able to establish global dependencies among different input tokens. Inspired by this, recent works [13,14] utilize transformer to implement 3D-HPE. These methods make an effort to explore the complex relationships between skeleton joints. Obvious performance improvements

* Corresponding author.
E-mail addresses: zhmchen@buaa.edu.cn (Z. Chen), daij@pcl.ac.cn (J. Dai), baijunxuan@cupes.edu.cn (J. Bai), pan_junjun@buaa.edu.cn (J. Pan).

https://doi.org/10.1016/j.patcog.2024.110446
Received 7 June 2022; Received in revised form 14 March 2024; Accepted 20 March 2024
Available online 24 March 2024

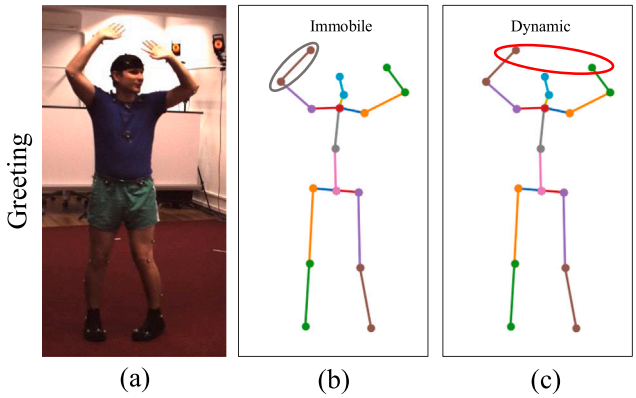

Fig. 1. Graph representation of human skeleton for *Greeting* action. **(a)** The original image. **(b)** The gray circle denotes the physical connections between *left wrist* and *elbow* joints. **(c)** The red circle refers to dynamic connections between *left wrist* and *right wrist*.

have been achieved by learning the global context information for updating and representation of joints. However, only applying the self-attention mechanism will weaken the graph structure expression of the human skeleton, which is usually used as a strong prior to estimating unusual postures. Considering this, some methods [15,16] attempt to integrate transformer and graph structure information via stacking GCN layers and transformer encoder. Nonetheless, they only consider the physical connections of human skeleton joints.

To address the above issues, we propose the Dynamic Graph Transformer (DGFormer) to learn the dynamic local and global relationships for 3D-HPE. The proposed DGFormer consists of a Transformer Encoder (TE), an immobile GCN module, and a dynamic GCN module, which are respectively responsible for constructing global long-range dependencies, local physical connections, and sparse dynamic relationships between skeleton joints. Particularly, the dynamic GCN is able to handle the pose-dependent correlations of joints. We evaluate the proposed DGFormer on available widely used datasets, i.e., Human3.6M [17] and MPI-INF-3DHP [18]. Experimental results demonstrate that our DGFormer achieves state-of-the-art performance. The **main contributions** of our method are listed as follows:

- We propose a novel framework, called Dynamic Graph Transformer (DGFormer), for 3D human pose estimation. Our method effectively takes advantage of the global and local dynamic relationships among human joints for performance improvements.
- Considering the prior information of the human skeleton, we formulate a new GCN block consisting of an immobile GCN and a dynamic GCN, which capture the multi-scale physical and sparse dynamic relationships of skeleton joints for diversified action poses, respectively.
- Experiments on two challenging datasets: Human3.6M and MPI-INF-3DHP, demonstrate that our method achieves state-of-the-art results compared with image-based methods. Our method has impressive generalization abilities through experiments on in-the-wild videos.

2. Related work

In this section, we first review the advanced 2D and 3D HPE methods. Next, we introduce the progressive GCN and vision transformer that are related to our work.

2.1. 2D human pose estimation

2D-HPE is a fundamental problem in computer vision, which detects and localizes 2D keypoints from 2D images or videos. With the progressive development of deep learning, 2D-HPE has achieved impressive improvements by utilizing CNNs. For example, the widely used OpenPose [19] is a real-time multiple-person pose detection approach that leverages the Part Affinity Fields (PAFs) to associate body parts with individuals in the image. CPN [9] presents a cascaded pyramid network with the globalNet localizing the relative simple keypoints, and the RefineNet aiming to handle the occluded and hard keypoints. HRNet [20] claims that high-resolution features are essential for position-sensitive vision tasks. Therefore, it maintains high-resolution representations throughout the whole process for estimating 2D human pose. In this paper, we concentrate on estimating 3D pose from 2D keypoints to reduce complexity, and the 2D pose can be obtained by existing state-of-the-art 2D-HPE methods in advance.

2.2. 3D human pose estimation

Recently, deep neural networks have become the prevalent technology for 3D-HPE. Currently, existing methods can be broadly classified into two categories: direct estimation [6,21] and 2D-to-3D lifting [5,7]. The former is end-to-end optimized, directly estimating 3D joint coordinates from original images. However, the computation is quite expensive, and models are hungry for labeled training samples. Thanks to the excellent performance of 2D human pose estimation [9,20], 2D-to-3D lifting methods receive increasing attention. For instance, Martinez et al. [7] design a simple and effective fully-connected residual network to estimate 3D pose based on 2D keypoints from a single frame. To maintain temporal consistency, works in [4,22] establish spatial–temporal correlations for input sequences. The 2D-to-3D lifting schema greatly reduces the task difficulty, and it can also utilize a large amount of available 2D pose estimation datasets. Therefore, in this paper, we adopt the 2D-to-3D lifting to conduct 3D-HPE.

2.3. Graph convolutional networks

GCNs are widely used to process graph-structured data. The human skeleton owns natural connections between joints, which is especially suitable for learning by GCNs. Numerous GCN-based variations have been proposed and devoted to addressing skeleton-based vision problems for the past few years. For instance, inspired by graph Laplacian, Kipf and welling [23] formulate graph convolutional networks by the Chebyshev approximation. ST-GCN [24] is the first method to adopt GCN for skeleton-based action recognition. Later, GCN becomes the prevailing technology in various fields. Zhong et al. [25] propose the Gating-Adjacency GCN network (GAGCN) to predict future motion movements given historical motion sequences. Korban et al. [26] utilize the graph structure to represent skeletal, posture, clothing, and facial information for age estimation. In 3D-HPE, SemGCN [10] leverages a semantic GCN to learn local and global contextual information. GAST-Net [27] resorts graph attention mechanisms to acquire kinematic constraints of the human skeleton by modeling local and global spatial information. GraphSH [11] formulates graph stacked hourglass networks to capture multi-scale and multi-level features on skeleton data. Nevertheless, the above methods learn skeleton representation using fixed physical connections. In contrast, we present a novel GCN block consisting of an immobile GCN and a dynamic GCN to learn high-order and dynamic dependencies with fixed and dynamic affinities between skeleton joints.

2.4. Vision transformer

Transformer [12] is originally proposed for natural language processing. It owns powerful global context modeling abilities because of the self-attention mechanism. Recently, several works utilize transformer for human pose estimation. For instance, PoseFormer [28] is a purely transformer-based approach for 3D-HPE in videos with

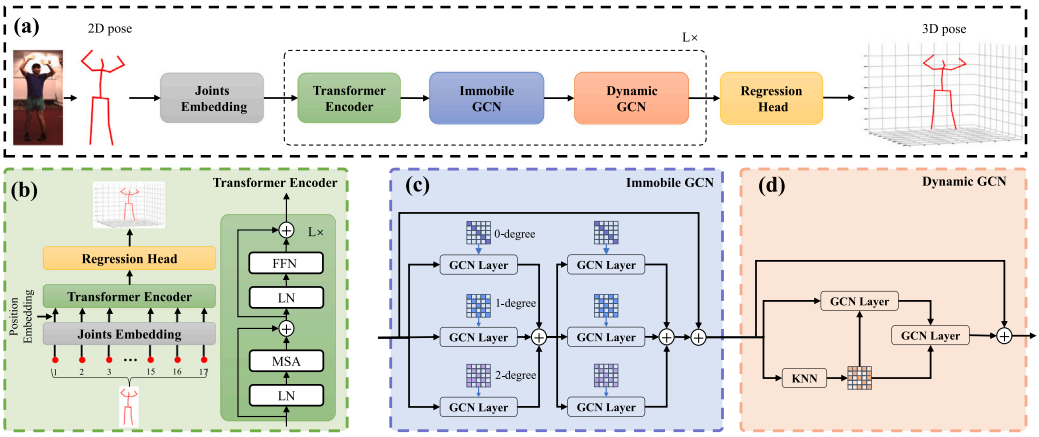

Fig. 2. **(a)** Overview of the proposed Dynamic Graph Transformer (DGFormer). **(b)** Transformer Encoder module. **(c)** Immobile GCN module. **(d)** Dynamic GCN module.

seq2frame solution. MixSTE [14] adopts the seq2seq solution to alternately model spatial correlations of joints and temporal dependencies among frames. MHFormer [13] learns spatio-temporal representations of multiple pose hypotheses to estimate 3D poses. These approaches estimate the 3D poses from video sequences. In addition, some works combine GCN and transformer to learn expressive features for structural data. Graphormer [29] explores three levels of graph encoding to enhance transformer modeling abilities. GPS [30] builds a common foundation for graph transformer that incorporates structural encodings with local message passing and global attention. In this paper, we focus on estimating 3D poses from a single image and incorporating the transformer with GCN to capture powerful and comprehensive relationships for skeleton joints.

3. Method

3.1. Preliminaries

In this work, we leverage the benefits of GCN and transformer for 3D-HPE. We first give a brief description of graph convolution operation (GCN) and the essential components of the transformer, including MSA (Multi-head Self-Attention) and FFN (Feed Forward Network).

GCN. Since the human skeleton is a natural graph structure, and it can be represented as $G = (V, E)$, where V and E are the node and edge sets of graph G. The graph adjacency matrix with J skeleton joints is denoted as $\mathbf{A} \in \mathbb{R}^{J\times J}$. Assuming the latent representation of an input pose data in layer l is expressed as $\mathbf{X}^{(l)} \in \mathbb{R}^{J\times d}$, where d refers to the embedding dimension. The graph convolution operation for the human skeleton can be represented as follows:

$$\mathbf{X}^{(l+1)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}^{(l)}\mathbf{\Theta}\right), \tag{1}$$

where $\sigma(\cdot)$ refers to the activation function, $\mathbf{\Theta} \in \mathbb{R}^{d\times d}$ denotes a learnable weight matrix, $\tilde{\mathbf{A}} = \mathbf{A}+\mathbf{I}$, $\tilde{\mathbf{D}}$ is the diagonal node degree matrix of $\tilde{\mathbf{A}}$, and $\mathbf{I}$ refers to the identity matrix.

MSA. The transformer can capture global long-range relationships, benefiting from the multi-head self-attention module. Following the standard procedures, the input $\mathbf{X}^{(l)} \in \mathbb{R}^{n\times d}$ is first mapped into $\mathbf{Q} \in \mathbb{R}^{n\times d}$, $\mathbf{K} \in \mathbb{R}^{n\times d}$, and $\mathbf{V} \in \mathbb{R}^{n\times d}$ with three linear layers, where n is the token number and d is the embedding dimension. We split the $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ for h heads, then calculate the scaled dot-product attention for head i:

$$\text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) = \text{softmax}\left(\frac{\mathbf{Q}_i\mathbf{K}_i^T}{\sqrt{d_k}}\right)\mathbf{V}_i. \tag{2}$$

where $\mathbf{Q}_i \in \mathbb{R}^{n\times d_k}$, $\mathbf{K}_i \in \mathbb{R}^{n\times d_k}$ and $\mathbf{V}_i \in \mathbb{R}^{n\times d_k}$ are the subsets of $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ for head i, and $d_k = d/h$. The h heads perform self-attention in parallel. We concatenate the outputs of h attention heads to obtain the updated data. The whole procedures can be formulated as:

$$\text{MSA}(\mathbf{X}^{(l)}) = \text{Concat}(\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_h)\mathbf{W}_o, \tag{3}$$

$$\mathbf{H}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i), i \in [1, \dots, h], \tag{4}$$

where $\mathbf{W}_o \in \mathbb{R}^{d\times d}$ is a parameter matrix that linearly transforms the outputs of all heads.

FFN. The FFN is applied after MSA with two linear layers for feature transformation and increase non-linearity. The formula is as follows:

$$\text{FFN}(\mathbf{X}^{(l)}) = \sigma(\mathbf{X}^{(l)}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{5}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d\times d_m}$ and $\mathbf{W}_2 \in \mathbb{R}^{d_m\times d}$ are weights of two linear layers respectively, and $\mathbf{b}_1$ and $\mathbf{b}_2$ are the bias terms.

3.2. Overview of the network

We adopt the 2D-to-3D lifting pattern for 3D human pose estimation. Given an input image, we first utilize the CPN detector [9] to obtain the 2D keypoints in advance. We illustrate the whole framework in Fig. 2. The proposed DGFormer mainly consists of the transformer encoder, immobile GCN, and dynamic GCN modules. We leverage the transformer encoder to learn global long-range context-dependency information of joints. To capture multi-scale local correlations of joints, the immobile GCN is formulated with the Chebyshev GCN [31]. As a result, the immobile GCN contains implicit higher-order information for extracting complex physical interactions among joints. The dynamic GCN is inspired by [32], which can adaptively integrate dynamic sparse contextual information of K-nearest neighbor joints according to different input poses. Although works in [16,33] combine the transformer and GCN to estimate the 3D poses, as well. However, both two methods do not consider the dynamic pose variations.

3.3. Transformer encoder

We advocate the transformer to learn long-range spatial correlations between joints. As shown in Fig. 2(b), given a 2D pose $\mathbf{X} \in \mathbb{R}^{J\times 2}$ with J joints, similar to NLP [12], each joint is regarded as a token. We map the 2D coordinates of each joint into a latent space through a linear transformation $\mathbf{W} \in \mathbb{R}^{2\times d}$. At the same time, a learnable spatial positional embedding $\mathbf{E}_p \in \mathbb{R}^{J\times d}$ is added to the latent representation of $\mathbf{X}$ to maintain spatial information. The formula is as follows:

$$\mathbf{X}^{(0)} = \mathbf{X}\mathbf{W} + \mathbf{E}_p. \tag{6}$$

$\mathbf{X}^{(0)}$ will be later fed into the transformer encoder, which can update the feature of each joint through integrating information from all joints.

The complete process of the transformer encoder can be formulated as follows:

$$\begin{aligned}\mathbf{X}'^{(l)} &= \mathbf{X}^{(l-1)} + \mathrm{MSA}(\mathrm{LN}(\mathbf{X}^{(l-1)})),\\ \mathbf{X}^{(l)} &= \mathbf{X}'^{(l)} + \mathrm{MLP}(\mathrm{LN}(\mathbf{X}'^{(l)})),\end{aligned} \tag{7}$$

where $\mathrm{LN}(\cdot)$ denotes the layer normalization operator, $l \in [1, \ldots, L]$ is the index of layers, and $\mathbf{X}^{(l)} \in \mathbb{R}^{J\times d}$ signifies the output of the transformer encoder for layer l. $\mathbf{X}^{(l)}$ will be later sent to the GCN block to reinforce local context information from both immobile GCN and dynamic GCN.

3.4. GCN block

The GCN Block in our network consists of an immobile GCN and a dynamic GCN, respectively responsible for capturing the natural physical connections and the sparse dynamic interactions. The two terms facilitate our method achieving noticeable performance improvements.

Immobile GCN. The immobile GCN in our network concentrates on leveraging the natural physical connection priors to address 3D pose estimation challenges. To capture multi-scale high-order context information, we employ Chebyshev polynomial as the convolutional kernel. We adopt the fixed adjacency matrix $\mathbf{A} \in \mathbb{R}^{J\times J}$ defined by physical connections of human skeleton to represent the edge between joint i and j:

$$\mathbf{A}(i,j) = \begin{cases} 1, & i \text{ and } j \text{ are connected in human skeleton,}\\ 0, & \text{otherwise.}\end{cases} \tag{8}$$

The Chebyshev graph convolution operation can be formulated as follows:

$$\mathbf{X}^{(l+1)} = \sigma\left(\sum_{m=0}^{M-1} \mathbf{T}_m(\tilde{\mathbf{L}})\mathbf{X}^{(l)}\mathbf{\Theta}_m\right), \tag{9}$$

where $\mathbf{T}_m(\cdot)$ denotes the Chebyshev polynomial of degree m evaluated with the normalized Laplacian $\tilde{\mathbf{L}} = 2\mathbf{L}/\lambda_{\max} - \mathbf{I} \in \mathbb{R}^{J\times J}$, and $\mathbf{L} = \mathbf{I} - \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$. $\lambda_{\max}$ is the max eigenvalue of $\mathbf{L}$, $\mathbf{I} \in \mathbb{R}^{J\times J}$ is the identity matrix, $\mathbf{\Theta}_m$ is a learnable parameter. The Chebyshev polynomial can be computed by the stable recurrence relation as follows:

$$\begin{aligned}&\mathbf{T}_0(\tilde{\mathbf{L}}) = \mathbf{I},\\ &\mathbf{T}_1(\tilde{\mathbf{L}}) = \tilde{\mathbf{L}},\\ &\mathbf{T}_m(\tilde{\mathbf{L}}) = 2\tilde{\mathbf{L}}\mathbf{T}_{m-1}(\tilde{\mathbf{L}}) - \mathbf{T}_{m-2}(\tilde{\mathbf{L}}).\end{aligned} \tag{10}$$

Compared with the vanilla GCN, the input $\mathbf{X}$ can integrate the features of m order neighbors for each joint. It should be noted that $\mathbf{A}$ is a fixed graph affinity representation based on physical connections of the human skeleton. However, the fixed affinity cannot establish non-physical connections for action poses that joints have high correlations but locate far away.

Dynamic GCN. To address the shortcomings of the immobile GCN, which only considers the physical connections of the human skeleton, we introduce the dynamic GCN. It aims to exploit the potential sparse long-range non-physical connections. Inspired by [32], we measure the correlation between joints by their distance, which can be calculated as:

$$R(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{Dist}(\mathbf{x}_i, \mathbf{x}_j), \tag{11}$$

where $\mathrm{Dist}(\cdot)$ is the distance between joint $\mathbf{x}_i$ and $\mathbf{x}_j$ in feature space. We adopt the Euclidean distance as the measure. Specifically, for each joint $\mathbf{x}_i$, we calculate its k nearest joints Ω_i, formulated as follows:

$$\Omega_i = \mathrm{KNN}\left(\mathbf{x}_i, \mathbf{x}_j, k\right), j \in [1, \ldots, J], \tag{12}$$

where KNN denotes the K-Nearest Neighbors algorithm. According to the set Ω, we construct an adaptive adjacency matrix $\mathbf{A}_{dym}$ to replace the immobile adjacency matrix $\mathbf{A}$:

$$\mathbf{A}_{dym}(i,j) = \begin{cases} 1, & j \in \Omega^i,\\ 0, & \text{otherwise.}\end{cases} \tag{13}$$

Then, the dynamic GCN operation is similar to the immobile GCN using Chebyshev polynomial. The formula is as follows:

$$\mathbf{X}^{(l+1)} = \sigma\left(\sum_{m=0}^{M-1} \mathbf{T}_m(\tilde{\mathbf{L}}_{dym})\mathbf{X}^{(l)}\mathbf{\Theta}_m\right), \tag{14}$$

where $\tilde{\mathbf{L}}_{dym}$ is calculated according to A_{dym}, referring to Eq. (9)

3.5. The proposed DGFormer

Our DGFormer leverages the transformer and the GCN block to establish global long-range and dynamic local context dependencies between human joints for 3D-HPE. As illustrated in Fig. 2, the whole network consists of three stages: joint embedding, feature encoding, and regression head.

Given 2D pose keypoints detected in advance, we first embed it into the latent space through a linear layer, in where the position embedding is added. Then, the position-aware features are processed sequentially via three modules: transformer encoder, immobile GCN, and dynamic GCN, to obtain pose representation. The transformer encoder is formulated with a vanilla transformer. In the immobile GCN module, we conduct m graph convolution operations with multi-scale adjacency matrices acquired by Chebyshev polynomial at each layer. We apply a summation operation for the m outputs. In the dynamic GCN module, the KNN method is employed to calculate the dynamic adjacency matrix. Both the GCN modules use the residual style with two layers. Finally, in the regression head, the intermediate features are projected into the 3D pose through a linear projection layer.

3.6. Loss function

We use Mean Per Joint Position Error (MPJPE) loss to train our network. MPJPE is applied to minimize the errors between the ground truth and predicted poses as:

$$\mathcal{L} = \frac{1}{NJ}\sum_{i=1}^{N}\sum_{j=1}^{J}\left\|\mathbf{Y}_{i,j} - \hat{\mathbf{Y}}_{i,j}\right\|_2, \tag{15}$$

where $\mathbf{Y}_{i,j}$ and $\hat{\mathbf{Y}}_{i,j}$ denote the ground truth and estimated 3D joint coordinates of jth joint for sample i, respectively.

4. Experiments

4.1. Datasets and evaluation metrics

In this paper, we conduct experiments on two widely used challenging 3D human pose estimation datasets, Human3.6M [17] and MPI-INF-3DHP [18] to evaluate the proposed DGFormer.

Human3.6M. Human3.6M [17] is the most widely used dataset for 3D single-person pose estimation. It contains 3.6 million images captured by the MoCap system in the indoor environment. In this dataset, 11 subjects are performing 15 actions from 4 different cameras. Following the previous works [4,34], five subjects (S1, S5, S6, S7, S8) are used for training, two subjects (S9, S11) are selected for testing. We adopt two evaluation protocols: protocol 1 is MPJPE between the ground truth and estimated 3D pose. protocol 2 is P-MPJPE which is MPJPE after rigid alignment.

MPI-INF-3DHP. MPI-INF-3DHP is a challenging 3D human pose estimation dataset consisting of constrained indoor and complex outdoor scenes. The dataset collects 8 subjects performing 8 action from 14 camera views. When implementing experiments on MPI-INF-3DHP, we direct predict 3D pose coordinates using the model trained on Human3.6M without fine-turning [35]. We utilize MPJPE, Percentage of Correct Keypoint (PCK) within the 150 mm range, and Area Under Curve (AUC) to evaluate this dataset.

Table 1
Experimental comparisons on the Human3.6M dataset with the detected 2D poses from CPN as network inputs. (†) represents that models use temporal information. The best results are highlighted in bold.

MPJPE (mm)(↓)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
SRNet [21] (ECCV'20)(†)	46.6	47.1	43.9	41.6	45.8	49.6	46.5	40.0	53.4	61.1	46.1	42.6	43.1	31.5	32.6	44.8
Anatomy3D [35] (TCSVT'21)(†)	42.1	43.8	41.0	43.8	46.1	53.5	42.4	43.1	53.9	60.5	45.7	42.1	46.2	32.2	33.8	44.6
PoseFormer [28] (ICCV'21)(†)	41.5	44.8	39.8	42.5	46.5	51.6	42.1	42.0	53.3	60.7	45.5	43.3	46.1	31.8	32.2	44.3
DG-Net [32] (TIP'21)(†)	41.5	46.6	41.0	44.3	47.1	54.1	44.2	42.5	54.9	58.8	46.9	43.1	46.9	32.6	35.6	45.3
PoseMoNet [3] (PR'22)(†)	42.7	45.0	40.5	43.4	46.4	51.4	46.0	40.7	52.3	51.1	44.2	44.1	43.4	38.1	38.3	44.3
MixSTE [14] (CVPR'22)(†)	37.6	40.9	37.3	39.7	42.3	49.9	40.1	39.8	51.7	55.0	42.1	39.8	41.0	27.9	27.9	40.9
SemGCN [10] (CVPR'19)	48.2	60.8	51.8	64.0	64.6	53.6	51.1	67.4	88.7	**57.7**	73.2	65.6	48.9	64.8	51.9	60.8
Sharma et al. [36] (ICCV'19)	48.6	54.5	54.2	55.7	62.2	72.0	50.5	54.3	70.0	78.3	58.1	55.4	61.4	45.2	49.7	58.0
LCN [34] (ICCV'19)	46.8	52.3	**44.7**	50.4	52.9	68.9	49.6	46.4	60.2	78.9	51.2	50.0	54.8	40.4	43.3	52.7
Liu et al. [37] (ECCV'20)	46.3	52.2	47.3	50.7	55.5	67.1	49.2	46.0	60.4	71.1	51.5	50.1	54.5	40.3	43.7	52.4
METRO [38] (CVPR'21)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	54.0
GraphSH [11] (CVPR'21)	**45.2**	49.9	47.5	50.9	54.9	66.1	**48.5**	46.3	59.7	71.5	51.4	48.6	53.9	39.9	44.1	51.9
GraFormer [16] (CVPR'22)	49.3	53.9	54.1	55.0	63.0	69.8	51.1	53.3	69.4	90.0	58.0	55.2	60.3	47.4	50.6	58.7
DGFormer (Ours, k=3)	45.8	**49.6**	46.2	**49.6**	51.4	58.7	48.9	46.2	56.6	65.1	50.9	**47.2**	53.2	38.8	41.5	50.0
DGFormer (Ours, k=7)	46.3	50.3	45.7	50.5	**50.8**	**57.5**	49.6	**46.0**	**55.8**	63.8	**50.9**	47.8	**53.0**	**38.7**	**41.3**	**49.8**
P-MPJPE (mm)(↓)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Anatomy3D [35] (TCSVT'21)(†)	33.1	35.3	33.4	35.9	36.1	41.7	32.8	33.3	42.6	49.4	37.0	32.7	36.5	25.5	27.9	35.6
PoseFormer [28] (ICCV'21)(†)	32.5	34.8	32.6	34.6	35.3	39.5	32.1	32.0	42.8	48.5	34.8	32.4	35.3	24.5	26.0	34.6
PoseMoNet [3] (PR'22)(†)	28.4	29.6	33.9	38.5	37.4	41.9	29.4	30.9	39.8	49.7	38.5	31.6	31.8	28.2	31.7	34.7
MixSTE [14] (CVPR'22)(†)	30.8	33.1	30.3	31.8	33.1	39.1	31.1	30.5	42.5	44.5	34.0	30.8	32.7	22.1	22.9	32.6
LCN [34] (ICCV'19)	36.9	41.6	38.0	41.0	41.9	51.1	38.2	37.6	49.1	62.1	43.1	39.9	43.5	32.2	37.0	42.2
Liu et al. [37] (ECCV'20)	35.9	40.0	38.0	41.5	42.5	51.4	37.8	36.0	48.6	56.6	41.8	38.3	42.7	31.7	36.2	41.2
DGFormer (Ours, k=3)	35.5	38.5	37.2	40.6	40.1	44.8	**37.5**	35.5	45.5	52.6	40.9	**36.0**	41.6	30.7	34.5	39.4
DGFormer (Ours, k=7)	**35.4**	**38.4**	**35.8**	**40.3**	**39.2**	**43.7**	37.6	**34.8**	**44.7**	**51.3**	**40.2**	36.1	**41.2**	**30.6**	**33.9**	**38.9**

Table 2
Experimental comparisons on the Human3.6M dataset with the ground truth 2D poses as network inputs. (†) represents that models use temporal information. The best results are highlighted in bold.

MPJPE (mm)(↓)	Dir.	Disc	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
SRNet [21] (ECCV'20)(†)	34.8	32.1	28.5	30.7	31.4	36.9	35.6	30.5	38.9	40.5	32.5	31.0	29.9	22.5	24.5	32.0
PoseFormer [28] (ICCV'21)(†)	30.0	33.6	29.9	31.0	30.2	33.3	34.8	31.4	37.8	38.6	31.7	31.5	29.0	23.3	23.1	31.3
Liu et al. [37] (ECCV'20)	36.8	40.3	33.0	36.3	37.5	45.0	39.7	34.9	40.3	47.7	37.4	38.5	38.6	29.6	32.0	37.8
GraphSH [11] (CVPR'21)	35.8	38.1	31.0	35.3	35.8	43.2	37.3	31.7	38.4	45.5	35.4	36.7	36.8	27.9	30.7	35.8
GraFormer [16] (CVPR'22)	32.0	38.0	30.4	34.4	34.7	43.3	35.2	31.4	38.0	46.2	34.2	35.7	36.1	27.4	30.6	35.2
PoseGTAC [33] (IJCAI'21)	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
DGFormer (Ours, k=3)	**31.3**	34.6	**28.1**	32.6	33.2	39.3	37.8	30.2	36.6	38.9	33.4	33.3	34.3	28.1	29.7	33.4
DGFormer (Ours, k=7)	31.5	**34.3**	28.2	**32.2**	**31.3**	**36.8**	**37.0**	**29.4**	**34.9**	**37.8**	**31.8**	**32.5**	**33.0**	**26.7**	**28.9**	**32.4**

4.2. Implementation details

We implement the proposed method in the Pytorch[1] platform with a single NVIDIA GTX 1080 Ti GPU. We apply pose flipping horizontally as data augmentation in training and testing phases. We set $L = 6$ for DGFormer, $h = 8$ for self-attention heads, and $d = 128$ for feature embedding dimension. The m order of affinity in immobile GCN is 3, and we report the results of k equals to 3 and 7 for KNN in dynamic GCN. We apply Adam optimizer with an initial learning rate of 0.0001 and a decay rate of 0.99 to train the model for 50 epochs. We use the cascaded pyramid network (CPN) [9] as the 2D pose detector for Human3.6M and ground truth 2D pose for MPI-INF-3DHP.

4.3. Comparison with state-of-the art

Result on Human3.6M. To validate the superior performance of the proposed DGFormer, we make comparisons with state-of-the-art methods on Human3.6M. The comparison results of using 2D poses detected by CPN as input are displayed in Table 1. The last column is the average error. It can be observed that our model outperforms other approaches (except methods using video sequences as inputs) under both MPJPE (49.8, top) and P-MPJPE (38.9, bottom) metrics, respectively. Compared with the very recent graph-transformer-based method, *i.e.*, GraFormer [16], DGFormer noticeably surpasses it by 8.9 (15%) in MPJPE. Moreover, DGFormer makes more accurate 3D pose predictions on complex actions, such as *Photo, Smoke, WalkDog*, which contain more challenges. The comparison results demonstrate that our method can achieve remarkable performance gains by exploiting adequate local and global context information among skeleton joints.

To avoid the interference of 2D keypoints estimated by 2D pose detectors, we further implement experiments on Human3.6M using ground truth 2D poses as input and compare with the most advanced approaches. We report the comparison results in Table 2. Our method obtains an average MPJPE of 32.4 under MPJPE protocol and achieves approximately 7.8% improvement compared with GraFormer [16]. Furthermore, DGFormer achieves the best score in 11 actions (except for methods using temporal information).

Result on MPI-INF-3DHP. In this paper, we also utilize the challenge MPI-INF-3DHP dataset to validate our model's generalization performance, since it contains both indoor and outdoor scenarios with diversified pose variations. We predict the 3D pose coordinates by directly applying the model trained on Human3.6M. Table 3 shows the quantitative results compared with other methods. Although we utilize the model trained on Human3.6M without fine-tuning or re-training post-process, our approach achieves the best performance on all evaluation metrics (PCK, AUC, and MPJPE). The results indicate that the proposed DGFormer can adapt to unseen datasets with great generalization ability.

Qualitative Results. We also display some visual 3D pose predictions of Human3.6M and MPI-INF-3DHP datasets. The comparison results

[1] https://pytorch.org/

Table 3
Experimental comparisons on the MPI-INF-3DHP dataset with the ground truth 2D poses as network inputs. (†) represents that the models use temporal information. The best results are highlighted in bold.

Method	PCK (%)(↑)	AUC (%)(↑)	MPJPE (mm)(↓)
Anatomy3D [35] (TCSVT'21)(†)	87.9	54.0	78.8
PoseFormer [28] (ICCV'21)(†)	88.6	56.4	77.1
MHFormer [13] (CVPR'21)(†)	88.6	56.4	77.1
MixSTE [14] (CVPR'22)(†)	94.4	66.5	54.9
Pavalakos et al. [4] (CVPR'18)	71.9	35.3	–
Ci et al. [39] (TPAMI'20)	74.0	36.7	–
GraphSH [11] (CVPR'21)	76.4	39.3	–
PoseGTAC [33] (IJCAI'21)	80.1	45.8	–
DGFormer (Ours, k=3)	84.4	52.5	83.9
DGFormer (Ours, k=7)	**85.5**	**53.6**	**80.4**

Table 4
Ablation studies on Human3.6M with ground truth 2D poses as network inputs. Baseline: Transformer Encoder. I-GCN: Immobile GCN. D-GCN: dynamic GCN.

Method	MPJPE (mm)(↓)	P-MPJPE (mm)(↓)
Transformer (Baseline)	36.9	30.0
Transformer + I-GCN	34.1	27.4
Transformer + D-GCN	35.7	27.9
Transformer + I-GCN + I-GCN	34.8	27.4
Transformer + D-GCN + D-GCN	35.1	27.4
Transformer + I-GCN + D-GCN (DGFormer)	32.4	25.6

of the GraFormer [16], the Transformer, our DGFormer, and ground truth on some challenging poses are shown in Fig. 3. The top and bottom three rows are the results on Human3.6M and MPI-INF-3DHP, respectively. We use green arrows to indicate where the predictions are different and blue circles to mark the corresponding positions on ground truths. It can be observed that for either constrained indoor or challenging outdoor actions, our method can predict more accurate 3D pose results than the GraFormer and Transformer models. Furthermore, our model can also make correct predictions when facing self-occlusions and depth ambiguities, which can be seen from row 1 and row 5 in column 4 of Fig. 3. The qualitative results further demonstrate the impressive prediction abilities of our approach.

4.4. Ablation studies

We conduct ablation experiments on Human3.6M dataset under MPJPE and P-MPJPE criterion to verify the effectiveness of each module in our model. To avoid being affected by estimation errors of different 2D keypoints detectors, we adopt the ground truth 2D poses as input. The transformer Encoder is regarded as the baseline model [12] as shown in Fig. 2(b). We report the results in Table 4. The details of modules as follows:

- Baseline: The baseline model contains 6 layers and 8 heads for the standard Transformer encoder. The embedded feature dimension is 128.
- Immobile GCN (I-GCN): As stated in Section 3.4, I-GCN captures multi-scale high-order context information based on the natural physical connection. We set the degree M of Chebyshev polynomials to 3.
- Dynamic GCN (D-GCN): As mentioned in , the structure of D-GCN is similar to I-GCN. The difference is that the affinity matrix is dynamic, calculated by KNN according to different action poses. The degree of Chebyshev polynomials $M = 3$. We set $K = 7$ for KNN method.

As can be observed in Table 4, the transformer is able to reduce prediction errors by 2.8 and 1.2 for MPJPE metric, 2.6 and 2.1 for P-MPJPE metric, respectively, when incorporating immobile GCN (I-GCN) or dynamic GCN (D-GCN). When considering the global context, local physical connections, and sparse long-range information simultaneously, the proposed DGFormer achieves the lowest prediction errors of 32.4 under the MPJPE protocol and 25.6 under the P-MPJPE protocol. To explore whether the performance gains arise from the increasing model capacity, we add two I-GCN and two D-GCN modules into the baseline model, making our DGFormer have the same parameters as the two variant models. We can observe that the prediction errors of both models are higher than our DGFormer. The results demonstrate that the global information, local topology clues, and sparse long-range relationships among joints are crucial to 3D human pose estimation. The performance gains prove the effectiveness and superiority of the proposed DGFormer.

Since different modules in our model capture different types of dependencies, we further visualize the joint dependencies in Fig. 4. We take the *Sitting Down* action of Subject $S11$ in Human3.6M dataset as an example. Specifically, **(a)** denotes the 2D image and predicted 3D pose. We take the *pelvis* joint as an example. The dotted line indicates that the two joints are related. The thicker the lines, the closer the dependencies between the skeleton joints. **(b)** represents the natural connections from I-GCN. **(c)** is the dynamic relationships computed on the input sample by D-GCN. **(d)** shows the global dependencies obtained by the transformer encoder. The results of **(b), (c), (d)** demonstrate that our method can effectively capture different types of interactions between joints.

Table 5
Parameter analysis experiments for different architecture parameters (depth, dimension) in DGFormer. The evaluation is performed on Human3.6M with MPJPE(mm) using CPN detected 2D poses as network inputs.

#	Depth (L)	Dimension (d)	MPJPE (mm)(↓)
1	4	32	54.3
2	6	32	53.6
3	8	32	53.4
4	4	64	50.4
5	6	64	51.7
6	8	64	51.0
7	4	128	50.0
8	**6**	**128**	**49.8**
9	8	128	51.2

4.5. Depth analysis and discussion

Architecture Parameters Analysis. Table 5 shows the performance under the MPJPE metric with various parameter combinations. Depth(L) represents the number of layers used in the transformer encoder, and Dimension(d) indicates the embedded feature dimension in the model. According to different embedding dimensions, we divide the configurations into three groups to verify the model performance under different configurations. As can be observed in Table 5, the 3D pose estimation error begin to increase as the depth increases and reduce as the embedding dimension increases. Therefore, we set $L = 6$ and $d = 128$ to achieve the best performance and balance model size.

***k* in dynamic GCN.** We conduct experiments on the Human 3.6M with the ground truth 2D poses as input to explicitly illustrate the effects of different neighbor joints k in dynamic GCN. We report the MPJPE and P-MPJPE results in Fig. 5. To ensure the rigor of the experiments, we use the mean results of multiple runs. We can observe that with the increase of k, both the MPJPE and P-MPJPE errors first decrease and then increase, indicating that too few or many neighbors for a joint may result in insufficient or noisy contextual information. In this paper, we simultaneously report experimental results of $k = 3$ and $k = 7$.

***m* in GCN.** We investigate the sensitivity of m in GCN mentioned in Section 3.4, where we apply Chebyshev polynomial as the convolution kernel. We fix $k = 7$, $L = 6$ and $d = 128$, when we change m in our DGFormer. For this experiment, we also utilize ground truth 2D poses as input on Human3.6M under MPJPE metric. m indicates that

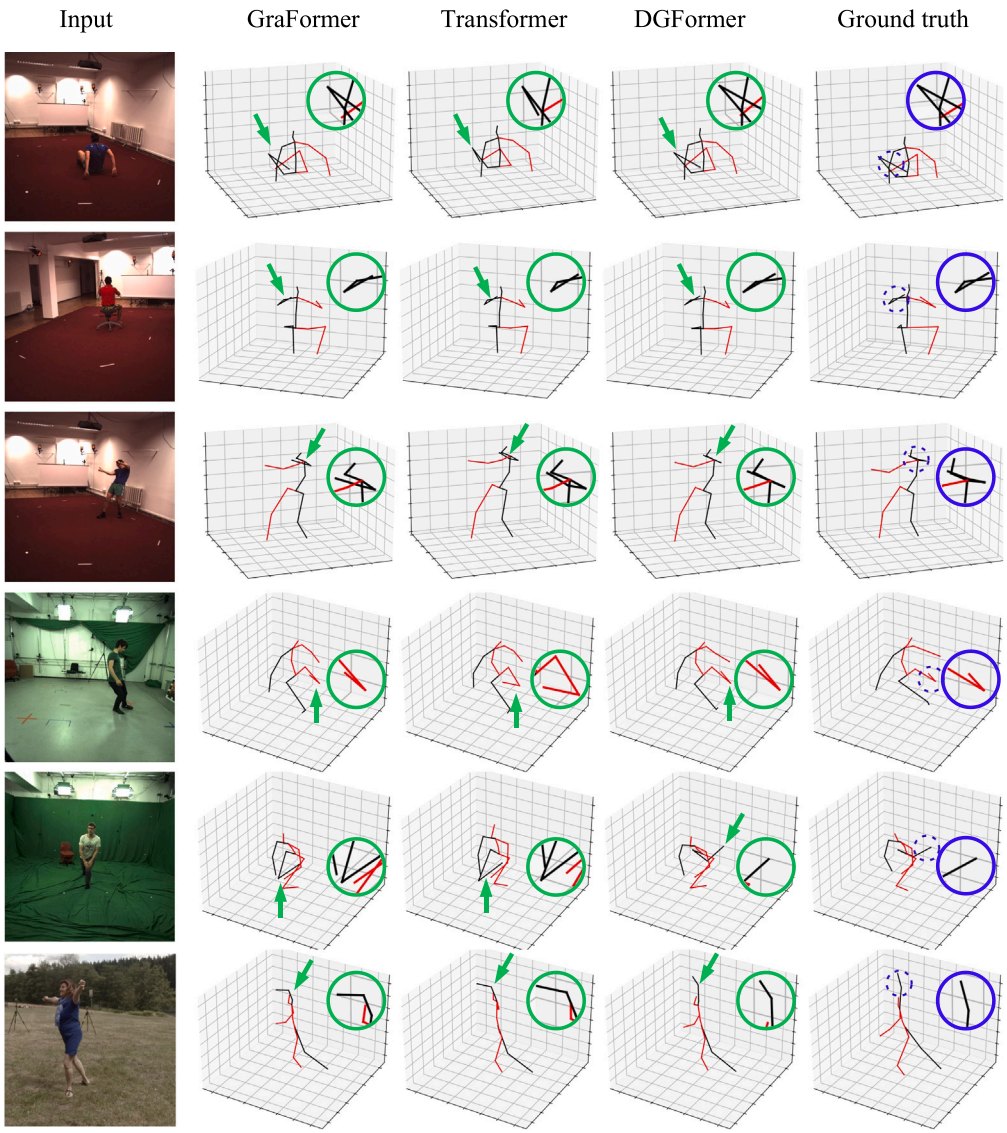

Fig. 3. Visualizations of 3D pose prediction on the Human3.6M and MPI-INF-3DHP datasets. The top three rows: results on Human3.6M. The bottom three rows: results on MPI-INF-3DHP.

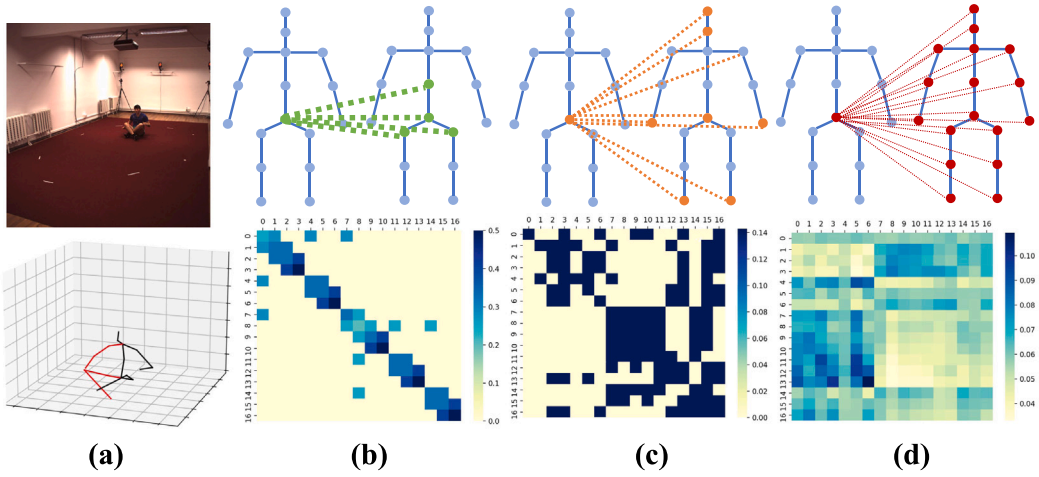

Fig. 4. Visualizations of learned joint dependency significance for the *SittingDown* action in Human3.6M test set S11.

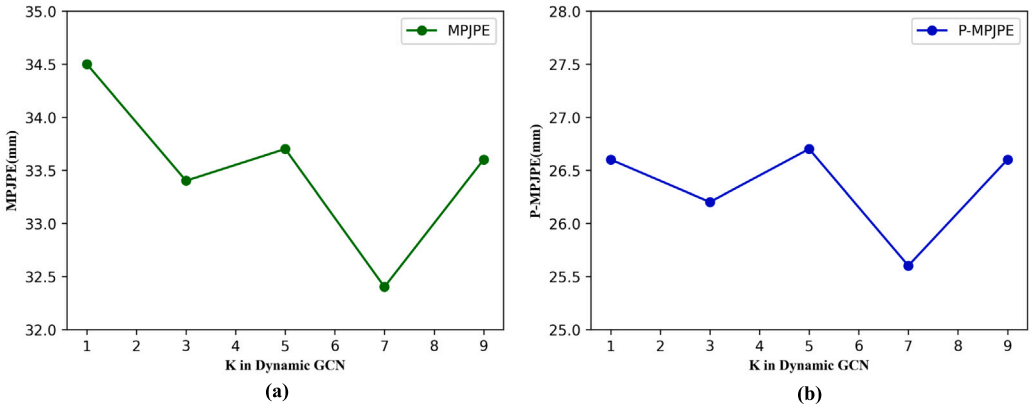

Fig. 5. Performance evaluations with different neighbor joints k in dynamic GCN on Human3.6M with ground truth 2D poses as inputs. (a) MPJPE, (b) P-MPJPE.

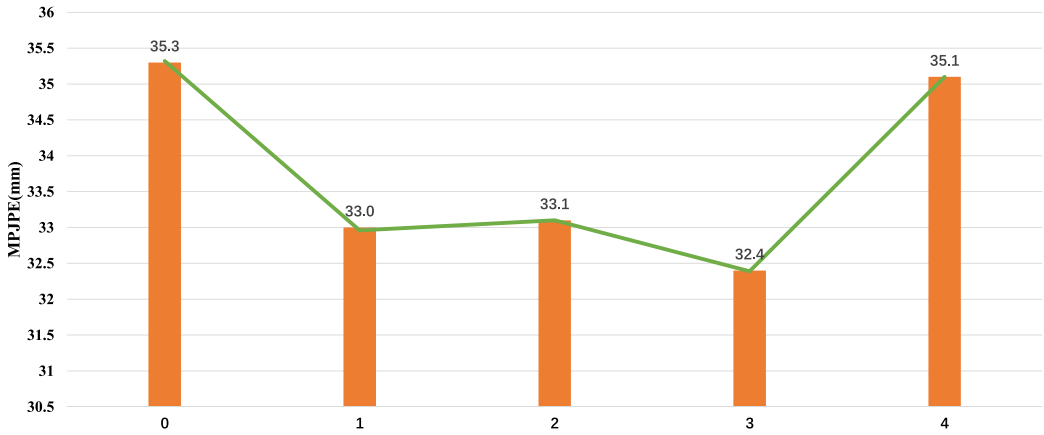

Fig. 6. Performance evaluations with different m (joint neighbors with m scales) in GCN. The evaluation performs on Human3.6M using ground truth 2D poses as inputs.

each joint has neighbors with m scales ranging from 0-order to m-order and $m = 0$ means each joint has only self-connection. As can be seen in Fig. 6, when m goes from 0 to 1, the estimation error decreases significantly. As expected, when we further increase m and set it as 3, the estimation error of DGFormer decreases from 35.3 to 32.4 with an 8.2% error reduction. It is unquestionable that the incorporation of multi-scale high-order context information can enhance model performance dramatically. However, when m continues to grow, the estimation error increases. The reason may be that too high-order neighbors will introduce some noisy contextual information. To balance efficiency and performance, we choose $m = 3$ in our DGFormer.

Discussion on model complexity. Finally, we make comparisons with state-of-the-art methods regarding model parameters, model performance, and computational time (frames per second (FPS) in the testing phase). The FPS for all the compared methods is calculated on a single NVIDIA GeForce RTX 3090 Ti GPU, with the codes and pre-trained models provided by the authors. The experiments are implemented on Human3.6M with the ground truth 2D poses as input. It can be observed from Table 6 although our method performs slightly worse than the video-based methods utilizing temporal information, *i.e.,* Anatomy3D [35], PoseFormer [28] and MHformer [13], our model has a much fewer model parameter. Nevertheless, it should be noted that our model obtains the best performance with a moderate model size compared with the single frame-based methods. In terms of computational time, the FPS of our method is much faster than the video-based methods. Even though the FPS is not the most promising compared with the single frame-based methods, our model's speed meets real-time requirements. The impressive prediction errors with small model parameters and fast inference speeds demonstrate the advantages of our model.

Table 6
Model complexity comparisons on Human3.6M using ground truth 2D poses as network inputs. (†) represents that the models use temporal information. FPS is computed on a single NVIDIA GeForce RTX 3090 Ti GPU.

Methods	Parameters (M)	MPJPE (mm)(↓)	FPS (↑)
Anatomy3D [35](†)	59.18	32.3	665
PoseFormer [28](†)	9.60	31.3	1940
MHFormer [13](†)	18.92	30.5	347
FC [7]	4.29	45.5	60 371
SemGCN [10]	0.43	43.8	15 676
Pre-agg [37]	4.22	37.8	–
GraphSH [11]	3.70	35.8	41 795
GraFormer [16]	0.65	35.2	59 512
DGFormer(Ours)	4.34	**32.4**	5544

4.6. Qualitative results on videos in-the-wild

Estimating 3D human poses from in-the-wild videos is challenging due to the complex environment and unknown camera parameters. Applying a pre-trained model to in-the-wild videos is a practical way to verify network generalization ability. Specifically, we first utilize

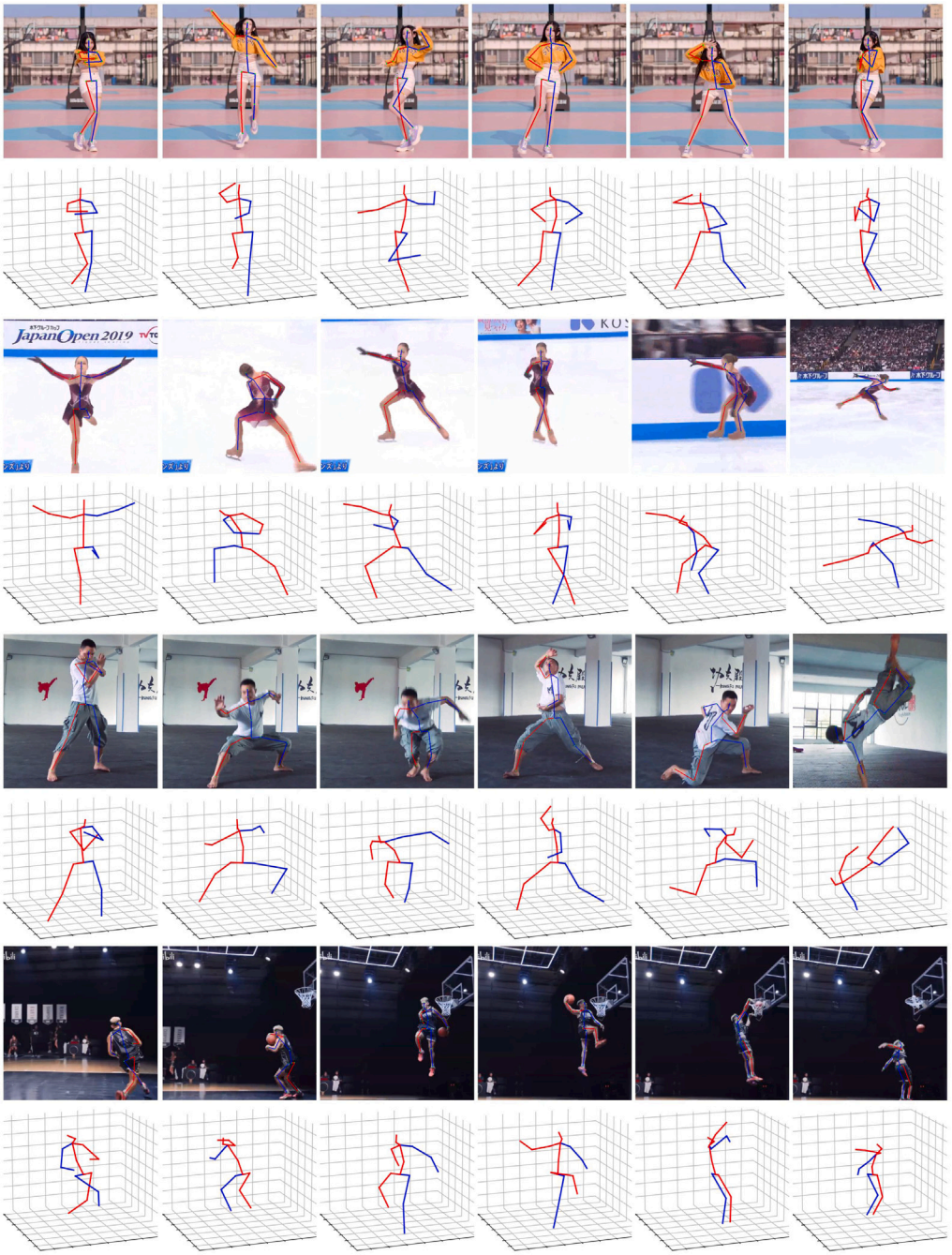

Fig. 7. Qualitative results of our method for in-the-wild videos.

YOLOv3 [40] to detect the person from videos, then employ HR-Net [20] as the 2D keypoints detector. At last, the pre-trained DG-Former on Human3.6M is used to estimate 3D human poses for videos on Bilibili.[2] We randomly choose challenging *dance*, *skating*, *martial arts*, and *dunking* videos as testing videos. As shown in Fig. 7, our method achieves plausible high fidelity results for in-the-wild videos, validating the superior generalization of our method.

5. Conclusion

In this paper, we propose the dynamic graph transformer network for 3D human pose estimation. The proposed model takes advantage of the transformer encoder, immobile GCN, and dynamic GCN modules to build the global long-range, sparse dynamic, and natural physical interactions of skeleton joints. Our method simultaneously leverages the global and local diversified context information for performance improvements. Comparative experiments on Human3.6M and MPI-INF-3DHP datasets demonstrate that our framework outperforms state-of-the-art image-based methods, and qualitative tests on in-the-wild

[2] https://www.bilibili.com/

videos also verify the superior generalization ability of our method. However, our method still performs slightly worse than the most advanced video-based methods with much fewer parameters since we do not utilize the temporal information of the skeleton sequence. In the future, we attempt to incorporate temporal information to further reduce the intrinsic depth arbitrariness, improve the robustness to occlusion and enhance the 3D pose prediction ability.

CRediT authorship contribution statement

Zhangmeng Chen: Writing – original draft, Visualization, Methodology. **Ju Dai:** Writing – review & editing, Writing – original draft, Funding acquisition. **Junxuan Bai:** Writing – review & editing. **Junjun Pan:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research is supported by National Science and Technology Major Project in China (No. No. 2022ZD0115902) and National Natural Science Foundation of China (No. 62102208.

References

[1] C. Han, X. Yu, C. Gao, N. Sang, Y. Yang, Single image based 3D human pose estimation via uncertainty learning, Pattern Recognit. 132 (2022) 108934.
[2] S. Du, Z. Yuan, T. Ikenaga, Kinematics-aware spatial-temporal feature transform for 3D human pose estimation, Pattern Recognit. 150 (2024) 110316.
[3] J. Yang, Y. Ma, X. Zuo, S. Wang, M. Gong, L. Cheng, 3D pose estimation and future motion prediction from 2D images, Pattern Recognit. 124 (2022) 108439.
[4] D. Pavllo, C. Feichtenhofer, D. Grangier, M. Auli, 3D human pose estimation in video with temporal convolutions and semi-supervised training, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 7753–7762.
[5] Z. Tang, Z. Qiu, Y. Hao, R. Hong, T. Yao, 3D human pose estimation with spatio-temporal criss-cross attention, in: IEEE Conference on Computer Vision and Pattern Recognition, 2023, pp. 4790–4799.
[6] G. Pavlakos, X. Zhou, K.G. Derpanis, K. Daniilidis, Coarse-to-fine volumetric prediction for single-image 3D human pose, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7025–7034.
[7] J. Martinez, R. Hossain, J. Romero, J.J. Little, A simple yet effective baseline for 3d human pose estimation, in: IEEE International Conference on Computer Vision, 2017, pp. 2640–2649.
[8] F. Moreno-Noguer, 3D human pose estimation from a single image via distance matrix regression, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2823–2832.
[9] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, J. Sun, Cascaded pyramid network for multi-person pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7103–7112.
[10] L. Zhao, X. Peng, Y. Tian, M. Kapadia, D.N. Metaxas, Semantic graph convolutional networks for 3d human pose regression, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3425–3435.
[11] T. Xu, W. Takano, Graph stacked hourglass networks for 3D human pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 16105–16114.
[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Conference on Neural Information Processing Systems, 2017, pp. 5998–6008.
[13] W. Li, H. Liu, H. Tang, P. Wang, L. Van Gool, Mhformer: Multi-hypothesis transformer for 3D human pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2022.
[14] J. Zhang, Z. Tu, J. Yang, Y. Chen, J. Yuan, MixSTE: Seq2seq mixed spatio-temporal encoder for 3D human pose estimation in video, in: IEEE Conference on Computer Vision and Pattern Recognition, 2022.
[15] K. Gong, J. Zhang, J. Feng, PoseAug: A differentiable pose augmentation framework for 3D human pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 8575–8584.
[16] W. Zhao, W. Wang, Y. Tian, GraFormer: Graph-oriented transformer for 3D pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 20438–20447.
[17] C. Ionescu, D. Papava, V. Olaru, C. Sminchisescu, Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2013) 1325–1339.
[18] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, C. Theobalt, Monocular 3d human pose estimation in the wild using improved cnn supervision, in: International Conference on 3D Vision, 2017, pp. 506–516.
[19] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7291–7299.
[20] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5693–5703.
[21] A. Zeng, X. Sun, F. Huang, M. Liu, Q. Xu, S. Lin, Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach, in: European Conference on Computer Vision, 2020, pp. 507–523.
[22] H.-W. Kim, G.-H. Lee, W.-J. Nam, K.-M. Jin, T.-K. Kang, G.-J. Yang, S.-W. Lee, MHCanonNet: Multi-Hypothesis Canonical lifting Network for self-supervised 3D human pose estimation in the wild video, Pattern Recognit. 145 (2024) 109908.
[23] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.
[24] S. Yan, Y. Xiong, D. Lin, Spatial temporal graph convolutional networks for skeleton-based action recognition, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 7444–7452.
[25] C. Zhong, L. Hu, Z. Zhang, Y. Ye, S. Xia, Spatio-temporal gating-adjacency GCN for human motion prediction, in: IEEE Conference on Computer Vision and Pattern Recognition, 2022, pp. 6447–6456.
[26] M. Korban, P. Youngs, S.T. Acton, TAA-GCN: A temporally aware adaptive graph convolutional network for age estimation, Pattern Recognit. 134 (2023) 109066.
[27] J. Liu, J. Rojas, Y. Li, Z. Liang, Y. Guan, N. Xi, H. Zhu, A graph attention spatio-temporal convolutional network for 3D human pose estimation in video, in: IEEE International Conference on Robotics and Automation, 2021, pp. 3374–3380.
[28] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, Z. Ding, 3D human pose estimation with spatial and temporal transformers, in: IEEE International Conference on Computer Vision, 2021, pp. 11636–11645.
[29] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, Di He, Y. Shen, T. Liu, Do transformers really perform badly for graph representation? in: Conference on Neural Information Processing Systems, 2021, pp. 28877–28888.
[30] L. Rampásek, M. Galkin, V.P. Dwivedi, A.T. Luu, G. Wolf, D. Beaini, Recipe for a general, powerful, scalable graph transformer, in: Conference on Neural Information Processing Systems, 2022.
[31] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, Adv. Neural Inf. Process. Syst. 29 (2016).
[32] J. Zhang, Y. Wang, Z. Zhou, T. Luan, Z. Wang, Y. Qiao, Learning dynamical human-joint affinity for 3D pose estimation in videos, IEEE Trans. Image Process. 30 (2021) 7914–7925.
[33] Y. Zhu, X. Xu, F. Shen, Y. Ji, L. Gao, H.T. Shen, PoseGTAC: Graph transformer encoder-decoder with atrous convolution for 3D human pose estimation, in: International Joint Conference on Artificial Intelligence, 2021, pp. 1359–1365.
[34] H. Ci, C. Wang, X. Ma, Y. Wang, Optimizing network structure for 3d human pose estimation, in: IEEE International Conference on Computer Vision, 2019, pp. 2262–2271.
[35] T. Chen, C. Fang, X. Shen, Y. Zhu, Z. Chen, J. Luo, Anatomy-aware 3D human pose estimation with bone-based pose decomposition, IEEE Trans. Circuits Syst. Video Technol. 32 (1) (2021) 198–209.
[36] S. Sharma, P.T. Varigonda, P. Bindal, A. Sharma, A. Jain, Monocular 3d human pose estimation by generation and ordinal ranking, in: IEEE International Conference on Computer Vision, 2019, pp. 2325–2334.
[37] K. Liu, R. Ding, Z. Zou, L. Wang, W. Tang, A comprehensive study of weight sharing in graph networks for 3d human pose estimation, in: European Conference on Computer Vision, 2020, pp. 318–334.
[38] K. Lin, L. Wang, Z. Liu, End-to-end human pose and mesh reconstruction with transformers, in: IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 1954–1963.
[39] H. Ci, X. Ma, C. Wang, Y. Wang, Locally connected network for monocular 3d human pose estimation, IEEE Trans. Pattern Anal. Mach. Intell. 44 (3) (2020) 1429–1442.
[40] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, 2018, arXiv preprint arXiv:1804.02767.

Zhangmeng Chen received the B.S. degree in School of Computer Science and Technology, Beijing Institute of Technology, Beijing, in 2012, the M.S degree in School of Computer and Information Technology, Beijing Jiaotong University, Beijing, in 2016. He is currently a Ph.D. candidate in School of Computer Science, Beihang University. His research interest is in deep learning, 3D human pose estimation, action recognition.

Ju Dai is currently a Research Assistant Fellow in Peng Cheng Laboratory (PCL), Shenzhen, China. She received both B.S. and M.S. degree in Electronic Engineering, China University of Geosciences (CUG), Wuhan, China, in 2011 and 2014, respectively, and the Ph.D. degree in Signal Processing in Dalian University of Technology (DUT), Dalian, China, in 2020. She worked in PCL as Postdoctoral Research Fellow from 2020 to 2022. Her research interests include human pose estimation, motion prediction, motion control, person re-identification and saliency detection.

Junxuan Bai is currently working for China Mobile Research Institute (CMRI) as a researcher. He received B.S. degree in Mathematics from Dalian Maritime University, Dalian, China, in 2012. He received M.S. degree and Ph.D. degree in Computer Science from Beihang University, Beijing, China, in 2015 and 2021. His research interests include computer animation, 3D human motion, and virtual surgery.

Junjun Pan is currently a professor in School of Computer Science, Beihang University. He received both B.S and M.S degree in School of Computer Science, Northwestern Polytechnical University, China. In 2006, he studied in National Centre for Computer Animation (NCCA), Bournemouth University, UK as Ph.D. candidate with full scholarship. In 2010, he received the Ph.D. degree and worked in NCCA as Postdoctoral Research Fellow. From 2012 to 2013, he worked as a Research Associate in Center for Modeling, Simulation and Imaging in Medicine, Rensselaer Polytechnic Institute, USA. In November 2013, he was appointed as Associate Professor in School of Computer Science, Beihang University, China. His research interests include virtual surgery and computer animation.