

# Learning Dynamical Human-Joint Affinity for 3D Pose Estimation in Videos

Junhao Zhang, Yali Wang, Zhipeng Zhou, Tianyu Luan, Zhe Wang and Yu Qiao, *Senior Member, IEEE*

**Abstract**—Graph Convolution Network (GCN) has been successfully used for 3D human pose estimation in videos. However, it is often built on the fixed human-joint affinity, according to human skeleton. This may reduce adaptation capacity of GCN to tackle complex spatio-temporal pose variations in videos. To alleviate this problem, we propose a novel Dynamical Graph Network (DG-Net), which can dynamically identify human-joint affinity, and estimate 3D pose by adaptively learning spatial/temporal joint relations from videos. Different from traditional graph convolution, we introduce Dynamical Spatial/Temporal Graph convolution (DSG/DTG) to discover spatial/temporal human-joint affinity for each video exemplar, depending on spatial distance/temporal movement similarity between human joints in this video. Hence, they can effectively understand which joints are spatially closer and/or have consistent motion, for reducing depth ambiguity and/or motion uncertainty when lifting 2D pose to 3D pose. We conduct extensive experiments on three popular benchmarks, e.g., Human3.6M, HumanEva-I, and MPI-INF-3DHP, where DG-Net outperforms a number of recent SOTA approaches with fewer input frames and model size.

**Index Terms**—3D Human Pose Estimation, Graph Convolution, Deep Learning

## I. INTRODUCTION

THIS paper attacks the problem of 3D pose estimation in videos, which is an important computer vision task, due to its wide applications in video surveillance, virtual reality, etc. With rapid development of deep learning, this task has achieved remarkable progresses [1]–[3]. In particular, recent studies [4]–[6] have shown that, graph convolution networks can effectively boost 3D pose estimation in videos, by learning spatio-temporal relations from 2D pose. However, spatial/temporal graph convolutions in these models are built upon the fixed human-joint affinity that is defined by human skeleton. As a result, it is often limited to tackle depth ambiguity and/or motion uncertainty of human joints in videos, without taking dynamical pose variations into account. We use two examples in Fig. 1 to illustrate this problem.

First, we discuss spatial graph convolution by a *SittingDown* case at Frame  $t$ . As shown in Fig. 1(a), *wrist* is connected with *elbow* in the human skeleton. Based on such fixed spatial affinity, spatial graph convolution leverages *elbow* as the main context to update *wrist*. But, *elbow* stays at a higher position that is relatively far from *wrist* in this *SittingDown* case. Consequently, when integrating context from *elbow*, spatial graph convolution would lift *wrist* to an unsatisfactory position that is higher than its ground truth. In fact, *wrist* is spatially

closer to *hip* and *pelvis* in this *SittingDown* case. To reduce depth ambiguity, it is necessary to further exploit such spatial affinity and learn context from these joints for estimating 3D position of *wrist*.

Second, we discuss temporal graph convolution by a *Walking* case from Frame  $t$  to  $t + 1$ . According to the fixed temporal affinity in Fig. 1(b), *ankle* at Frame  $t$  is connected with the same joint *ankle* at Frame  $t + 1$ . In this case, temporal graph convolution leverages *ankle* at Frame  $t + 1$  as temporal context for estimating *ankle* at Frame  $t$ . But, *ankle* at Frame  $t + 1$  is moving forward. Only integrating such context would mislead temporal graph convolution to estimate *ankle* at Frame  $t$  to be forward. To capture how *ankle* moves correctly, it would be better to use other joints with similar motion as contextual guidance. In fact, in this *Walking* case, *ankle* keeps the consistent movement with *wrist* and *shoulder* from Frame  $t$  to Frame  $t + 1$ . To reduce motion uncertainty, it is necessary to further enhance such temporal affinity, and integrate contexts from *wrist* and *shoulder* at Frame  $t + 1$  for estimating 3D position of *ankle* at  $t$ .

Both cases show that, the fixed human-joint affinity often lacks adaptation capacity to describe the personalized pose dynamics in videos. To alleviate this difficulty, we propose a novel Dynamical Graph Network (DG-Net), which can dynamically adjust spatial/temporal human-joint affinity in videos, and adaptively integrate joint-dependent context for accurate 3D pose estimation. More specifically, we first design two dynamical graph convolution operations, i.e., Dynamical Spatial Graph (DSG) and Dynamical Temporal Graph (DTG) convolution. DSG/DTG can dynamically construct human-joint affinity, according to similarity of spatial distance/temporal movement between joints. Hence, they can capture robust human-joint relations to tackle pose variations in different videos. Second, we embed DSG and DTG into a dynamical graph convolution block, and cascade a number of such blocks progressively to build up DG-Net. Via adding 3D pose supervision on individual blocks and fusion of different blocks, our DG-Net can regularize the cooperative power of spatial and temporal joint relations in each block, and integrate complementary features in different blocks to boost 3D pose estimation in videos. Finally, we perform extensive experiments on widely-used human pose benchmarks, i.e., Human3.6M, HumanEva-I, and MPI-INF-3DHP. Our DG-Net outperforms a number of recent SOTA approaches.

## II. RELATED WORK

**3D Pose Estimation.** 3D pose estimation has been mainly driven by deep neural networks. Currently, most approaches

J. Zhang, Y. Wang, T. Luan, and Y. Qiao are with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China. Z. Wang is with the University of California, Irvine. E-mail: yu.qiao@siat.ac.cn

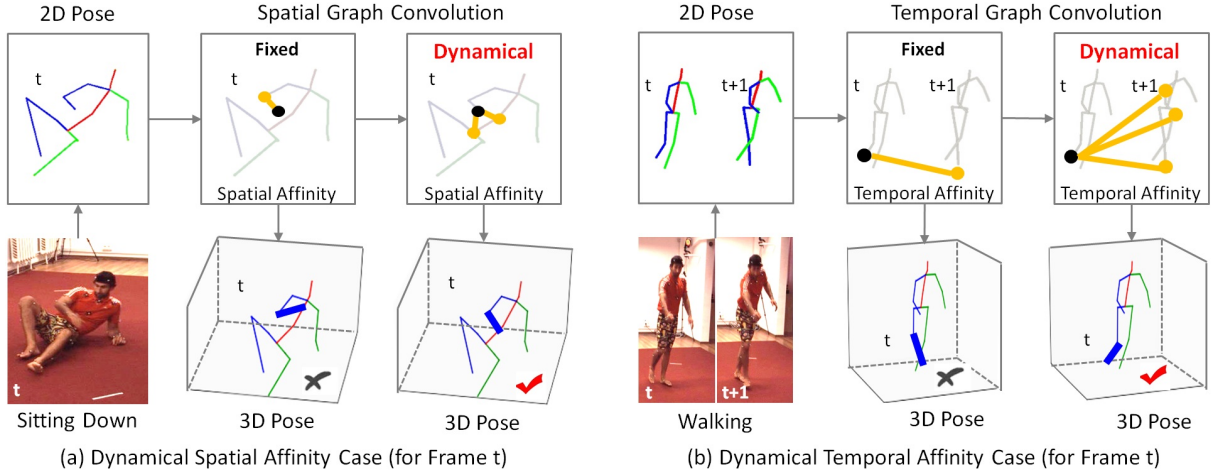


Fig. 1: Our motivations. The fixed human-joint affinity often limits adaptation capacity of spatial/temporal graph convolution to tackle complex pose variations in videos. To alleviate such difficulty, we propose to learn spatial/temporal human-joint affinity dynamically, according to spatial/temporal similarity between joints. For example, *wrist* is spatially closer to *hip* and *pelvis* in the *SittingDown* case of Fig. 1(a). Hence, we should further enhance such spatial affinity to estimate *wrist* at Frame  $t$ . Similarly, *ankle* keeps the consistent movement with *wrist* and *shoulder* in the *Walking* case. Hence, we should further enhance such temporal affinity to estimate *ankle* at Frame  $t$ . More explanations can be found in the introduction.

can be categorized into two frameworks. One framework is to directly estimate 3D joint locations from images [7]–[10]. A popular solution in this framework is based on the volumetric representation of human pose [8]. However, this method requires the non-differentiable post-processing step and introduces the quantization error [9]. To alleviate such difficulty, several extensions have been proposed by integral regression [9], ordinal depth supervision [10], etc. Another framework is to first estimate 2D pose and then lift 2D pose to 3D pose [4]–[6], [11]–[13], due to the fast development of 2D pose estimation [14], [15]. For instance, [11] introduces a simple but effective residual fully connected network to regress 3D pose from 2D pose. [16] formulates the problem as a 2D-to-3D distance matrix regression, which is often robust to missing observations. [2], [3], [17] encodes the temporal information in their lifting network, in order to tackle video 3D pose estimation with temporal smoothness. However, these approaches are often limited to learn distinct spatio-temporal relations between human joints in different actions, i.e., an important clue of pose ambiguity reduction. Alternatively, our DG-Net can capture rich relations by learning human-joint affinity dynamically, which effectively boosts 3D pose estimation in videos.

**Graph Convolution Networks.** Graph Convolution Network (GCN) is widely used to model graph-structured data [18]–[22]. Recently, it has been adopted for human-centric computer vision tasks [4], [23], by treating human skeleton as a specific graph. In particular, it is used to lift 2D input pose for 3D pose estimation, by learning human-joint relations [4]–[6], [24]. For instance, [4] introduces a semantic GCN, which learns channel-wise edge weights for enhancing relations between joints. [6] proposes a local-to-global GCN network, in order to learn multi-scale pose relations for 3D estimation. However, these approaches are built upon a fixed spatial/temporal graph affinity, i.e., the connections between

joints are based on the fixed human skeleton. Hence, they are limited to capture complex pose variations in videos. On the contrary, our DG-Net can adjust human-joint affinity adaptively to enhance 3D pose estimation, depending on the personalized poses of different actors in the video.

### III. METHOD

In this section, we first analyze spatial/temporal graph convolution with fixed human-joint affinity, and explain how to design our dynamical human-joint affinity. Then, we integrate Dynamical Spatial/Temporal Graph convolution (DSG/DTG) to build up our Dynamical Graph Network (DG-Net) for 3D human pose estimation in videos.

#### A. Dynamical Spatial Graph Convolution

For 3D pose estimation in videos, spatial graph is to describe spatial relations between human joints at each frame. Without loss of generality, we denote it as  $\mathcal{G}^S = (\mathcal{V}^S, \mathcal{E}^S)$ , where  $\mathcal{V}^S$  and  $\mathcal{E}^S$  are respectively the node and edge sets. Each node in  $\mathcal{V}^S$  refers to a human joint at a frame, while each edge in  $\mathcal{E}^S$  refers to the connection between two joints.

**Fixed Spatial Graph (FSG) Convolution.** For frame  $t$ , the node set  $\mathcal{V}^S$  corresponds to a feature matrix of human joints in this frame, i.e.,  $\mathbf{X}_t = [\mathbf{x}_t^1, \mathbf{x}_t^2, \dots, \mathbf{x}_t^N] \in \mathbb{R}^{N \times C_x}$ , where  $\mathbf{x}_t^i \in \mathbb{R}^{C_x}$  is the feature vector of the  $i$ -th human joint and  $N$  is the number of joints. Furthermore, the edge set  $\mathcal{E}^S$  is represented by an affinity matrix  $\mathbf{A} \in \mathbb{R}^{N \times N}$ . Traditionally, the edge between joint  $i$  and  $j$  is defined by human skeleton. Hence, the affinity matrix  $\mathbf{A}$  is fixed, i.e.,

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

Based on such fixed affinity, spatial graph convolution performs message passing to update the node feature  $\mathbf{X}_t$  as  $\mathbf{Y}_t \in \mathbb{R}^{N \times C_y}$ ,

$$\mathbf{Y}_t = \sigma(\mathbf{A}\mathbf{X}_t\Theta), \quad (2)$$

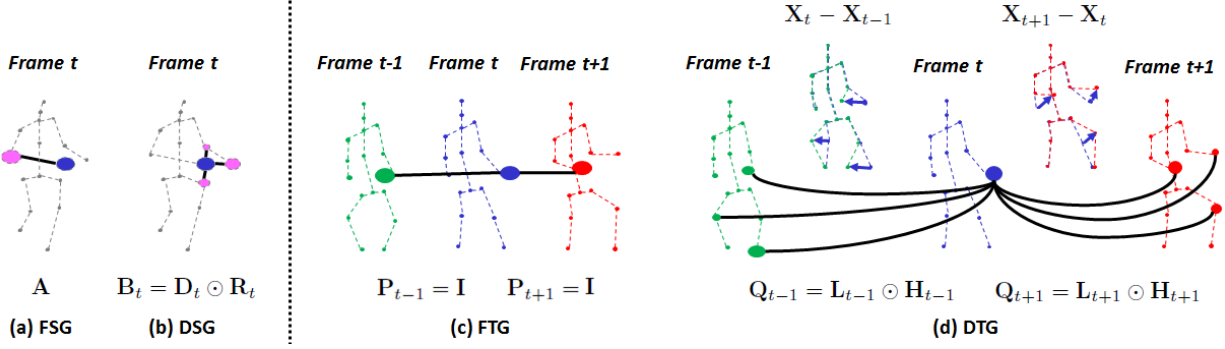


Fig. 2: Graph Illustration. (a) Fixed Spatial Graph (FSG). Graph affinity  $A$  is constructed according to fixed spatial structure of human skeleton. (b) Dynamical Spatial Graph (DSG). Graph affinity  $B_t$  is constructed by finding connections of spatial neighbors  $D_t$  and weighting importance of these neighbors  $R_t$ . (c) Fixed Temporal Graph (FTG). Forward and backward graph affinity  $P_{t+1}$ ,  $P_{t+1}$  are constructed according to fixed temporal structure of human skeleton. (d) Dynamical Temporal Graph (DTG). Forward and backward graph affinity  $Q_{t+1}$ ,  $Q_{t+1}$  are constructed by finding connections of temporal neighbors  $L_{t+1}$ ,  $L_{t-1}$  and weighting importance of these neighbors  $H_{t+1}$ ,  $H_{t-1}$ . The black line denotes the connection between joints. The size of circle in DSG and DTG refers to the importance of connected neighbors. More details can be found in Section III.

where  $\Theta \in \mathbb{R}^{C_x \times C_y}$  is the parameter matrix, and  $\sigma$  is a nonlinear activation function, e.g., ReLU. Note that, since  $A$  is based on the fixed human skeleton, Eq.(2) updates the feature of each human joint by integrating spatial context only from its physically-connected joints. However, a joint is not necessarily close to its physically-connected joints in different video frames, e.g., for the *SittingDown* case in the introduction, *wrist* is physically connected with *elbow*, while its location is closer to and influenced by *hip* and *pelvis*. Only integrating context from *elbow* may lead to unsatisfactory estimation of *wrist*.

**Dynamical Spatial Graph (DSG) Convolution.** To tackle the problem caused by the fixed affinity, we propose to adaptively construct edges between joints for each frame. Specifically, for a human joint  $i$  at frame  $t$ , we find a set of its  $K$  nearest joints  $\Omega_t^i$ , according to the feature matrix of human joints  $X_t$  in this frame,

$$\Omega_t^i = \text{KNN}(x_t^i, X_t, K), \quad (3)$$

where KNN refers to the  $K$ -Nearest-Neighbor algorithm, and we compute Euclidean distance between  $x_t^i$  and  $X_t$  to identify  $\Omega_t^i$ . When joint  $j$  belongs to  $\Omega_t^i$ , there is an edge between joint  $i$  and  $j$ . Formally, we can obtain a dynamical affinity matrix  $D_t$ ,

$$D_t(i, j) = \begin{cases} 1, & j \in \Omega_t^i \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Moreover, for joint  $i$ , the importance of its neighbor joints in  $\Omega_t^i$  can be different. To take it into account, we introduce a concise weighting mechanism to represent the importance score of joint  $j \in \Omega_t^i$ ,

$$R_t(i, j) = \gamma([x_t^i, x_t^j]), \quad (5)$$

where  $\gamma(\cdot)$  is a nonlinear mapping with the concatenated input of  $[x_t^i, x_t^j]$ . In our experiment, a fully-connected layer works well for  $\gamma(\cdot)$ . Finally, we obtain a weighted affinity via element-wise multiplication  $\odot$ ,

$$B_t = D_t \odot R_t. \quad (6)$$

This leads to our dynamical spatial graph convolution,

$$Y_t = \sigma(B_t X_t \Phi), \quad (7)$$

where  $\Phi \in \mathbb{R}^{C_x \times C_y}$  is a parameter matrix. Compared to the fixed and physical  $A$  in FSG (Eq. 2),  $B_t$  in DSG is adaptively generated from human joint feature  $X_t$ . Hence, it makes DSG robust to pose variations in different frames, by dynamically finding and weighting the important spatial neighbors for each joint.

### B. Dynamical Temporal Graph Convolution

Spatial graph convolution mainly focuses on learning spatial relations between human joints in each frame, while it ignores temporal relations between human joints in different frames. To bridge this gap, it is necessary to introduce temporal graph convolution. Specifically, we denote temporal graph as  $\mathcal{G}^T = (\mathcal{V}^T, \mathcal{E}^T)$ , where  $\mathcal{V}^T$  and  $\mathcal{E}^T$  are respectively the node and edge sets. Without loss of generality, we describe temporal graph at frame  $t$  in the following. First, the node set  $\mathcal{V}^T$  consists of human joints in three adjacent frames,  $t-1$ ,  $t$  and  $t+1$ . This is mainly because we would like to leverage both *forward* and *backward* temporal context for frame  $t$ . In this case, the node set involves three matrices of human joint features, i.e.,  $X_{t-1}$ ,  $X_t$  and  $X_{t+1}$ . Second, the edge set  $\mathcal{E}^T$  consists of human-joint connections between two adjacent frames. Hence, it refers to two affinity matrices, i.e., a *forward* affinity matrix that describes connections between frame  $t$  and  $t+1$ , and a *backward* affinity matrix that describes connections between frame  $t$  and  $t-1$ .

**Fixed Temporal Graph (FTG) Convolution.** In the traditional temporal graph, each human joint at frame  $t$  is assumed to be connected with the same joint at frame  $t+1$  and  $t-1$ . As a result, the forward and backward affinity matrices  $P_{t+1}$ ,  $P_{t-1} \in \mathbb{R}^{N \times N}$  are fixed and identical,

$$P_{t+1} = P_{t-1} = I, \quad (8)$$

i.e., when joint  $i$  at  $t$  and joint  $j$  at  $t+1$  are corresponding to the same joint,  $P_{t+1}(i, j) = 1$ . Otherwise,  $P_{t+1}(i, j) =$

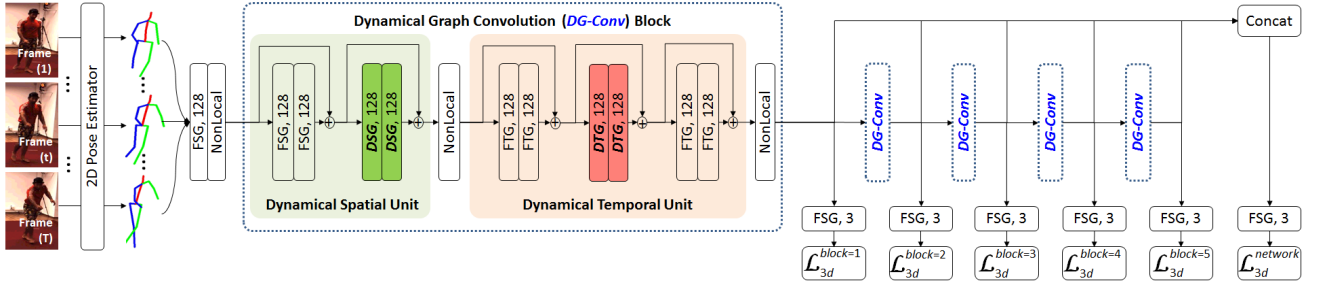


Fig. 3: Our DG-Net Architecture. We design a dynamical graph convolution (DG-Conv) block to lift 2D pose to 3D pose. Additionally, we supervise DG-Net in a multi-level manner, by leveraging the cooperation power of different DG-Conv blocks. More details can be found in Section III-C.

0. It is the same case for  $\mathbf{P}_{t-1}(i, j)$ . Based on such affinity matrices, temporal graph convolution is actually reduced as the traditional temporal convolution,

$$\begin{aligned} \mathbf{Y}_t &= \sigma(\mathbf{X}_t \mathbf{W}_x + \mathbf{P}_{t+1} \mathbf{X}_{t+1} \mathbf{W}_f + \mathbf{P}_{t-1} \mathbf{X}_{t-1} \mathbf{W}_b) \\ &= \sigma(\mathbf{X}_t \mathbf{W}_x + \mathbf{X}_{t+1} \mathbf{W}_f + \mathbf{X}_{t-1} \mathbf{W}_b), \end{aligned} \quad (9)$$

where  $\mathbf{W}_x, \mathbf{W}_f, \mathbf{W}_b \in \mathbb{R}^{C_x \times C_y}$  are the parameter matrices. One can see that, Eq. (9) updates the feature of each human joint at  $t$ , only by the features of the same joint at  $t-1$  and  $t+1$ . However, such temporal context is often insufficient to reflect how this joint moves, e.g., for *Walking* in the introduction, *ankle* actually keeps the consistent movement with *wrist* and *shoulder* from  $t$  to  $t+1$ . If we integrate context only from *ankle* at  $t+1$ , temporal graph convolution tends to mistakenly estimate *ankle* at  $t$  to be forward.

**Dynamical Temporal Graph (DTG) Convolution.** To deal with such problem in the temporal graph, we propose to dynamically discover joint edges between two adjacent frames. As mentioned in the introduction, for a human joint at  $t$ , other joints with similar movement (forward and/or backward) are the important temporal context to reduce motion uncertainty of this joint. Hence, we leverage the difference of joint features as guidance, and find these contextual joints from frame  $t+1$  and  $t-1$ . Specifically, for joint  $i$  at frame  $t$ , we use KNN to find a set of  $K$  related joints at frame  $t+1$  with similar forward motion,

$$\mathcal{U}_{t+1}^i = \text{KNN}(\mathbf{x}_{t+1}^i - \mathbf{x}_t^i, \mathbf{X}_{t+1} - \mathbf{X}_t, K), \quad (10)$$

where we compute Euclidean distance of feature difference between this joint  $\mathbf{x}_{t+1}^i - \mathbf{x}_t^i$  and all the joints in  $\mathbf{X}_{t+1} - \mathbf{X}_t$ , in order to identify the forward set  $\mathcal{U}_{t+1}^i$ . Similarly, we can find a set of  $K$  joints at frame  $t-1$  with similar backward motion,  $\mathcal{U}_{t-1}^i = \text{KNN}(\mathbf{x}_t^i - \mathbf{x}_{t-1}^i, \mathbf{X}_t - \mathbf{X}_{t-1}, K)$ . Based on  $\mathcal{U}_{t+1}^i$  and  $\mathcal{U}_{t-1}^i$ , one can construct the associated affinity matrices that change over time, e.g., when joint  $j$  at frame  $t+1$  belongs to the forward set  $\mathcal{U}_{t+1}^i$ , there is an edge between joint  $i$  at frame  $t$  and joint  $j$  at frame  $t+1$ ,

$$\mathbf{L}_{t+1}(i, j) = \begin{cases} 1, & j \in \mathcal{U}_{t+1}^i \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

The backward case is similar, i.e.,  $\mathbf{L}_{t-1}(i, j) = 1$  when  $j \in \mathcal{U}_{t-1}^i$ . Otherwise,  $\mathbf{L}_{t-1}(i, j) = 0$ . Moreover, for joint  $i$  at  $t$ , the importance of joints in the forward set (or the backward

set) can be different. Hence, like our dynamical spatial graph convolution before, we introduce a weighting mechanism, e.g., for joint  $i$  at  $t$ , we compute the importance score of joint  $j$  in the forward set  $\mathcal{U}_{t+1}^i$ ,

$$\mathbf{H}_{t+1}(i, j) = \alpha([\mathbf{x}_t^i, \mathbf{x}_{t+1}^j]). \quad (12)$$

Similarly, we introduce a weighting mechanism for joint  $j$  in the backward set  $\mathcal{U}_{t-1}^i$ , i.e.,  $\mathbf{H}_{t-1}(i, j) = \beta([\mathbf{x}_t^i, \mathbf{x}_{t-1}^j])$ . In our experiment, a fully-connected layer works well for  $\alpha(\cdot)$  and  $\beta(\cdot)$  respectively. Subsequently, we obtain the weighted affinity matrices for both forward and backward cases,

$$\mathbf{Q}_{t+1} = \mathbf{L}_{t+1} \odot \mathbf{H}_{t+1}, \quad \mathbf{Q}_{t-1} = \mathbf{L}_{t-1} \odot \mathbf{H}_{t-1}. \quad (13)$$

This leads to our dynamical temporal graph convolution,

$$\mathbf{Y}_t = \sigma(\mathbf{X}_t \mathbf{U}_x + \mathbf{Q}_{t+1} \mathbf{X}_{t+1} \mathbf{U}_f + \mathbf{Q}_{t-1} \mathbf{X}_{t-1} \mathbf{U}_b), \quad (14)$$

where  $\mathbf{U}_x, \mathbf{U}_f, \mathbf{U}_b \in \mathbb{R}^{C_x \times C_y}$  are the parameter matrices. Compared to FTG in Eq. (9), our DTG contains two time-varying affinity matrices  $\mathbf{Q}_{t+1}$  and  $\mathbf{Q}_{t-1}$ . For each joint at  $t$ , these matrices can effectively discover and weight its important temporal neighbors, according to movement trends. It allows our DTG to reduce motion uncertainty when estimating this joint.

### C. Dynamical Graph Network

In this section, we integrate dynamical spatial and temporal graph convolution to build a Dynamical Graph Network (DG-Net) for 3D pose estimation in videos. Specifically, recent studies [4], [11] have demonstrated that 3D pose can be lifted from 2D pose. Hence, we follow this concise style to design DG-Net. As shown in Fig. 3, we first use 2D pose estimator to predict 2D pose in each sampled frame, and then design a dynamical graph convolution block to estimate 3D pose by learning spatio-temporal relations of human joints.

**Dynamical Graph Convolution (DG-Conv) Block.** Since spatio-temporal factorization has been widely used for video learning [23], [31], we build a DG-Conv block with two dynamical units. First, dynamical spatial unit learns spatial pose relations in each frame. Then, dynamical temporal unit learns temporal pose relations between different frames. Moreover, as shown in Fig. 3, each dynamical unit is mixed with both fixed and dynamical graph convolution operations, in order to take advantage of their complementary characteristics for



Protocol1	$T$	Direct.	Discuss	Eat	Greet	Phone	Photo	Pose	Purch.	Sitting	SitD.	Smoke	Wait	WalkD	Walk.	WalkT.	Avg
Martinez et al. [11]	1	51.8	56.2	58.	59.0	69.5	78.4	55.2	58.1	74.0	94.6	62.3	59.1	65.1	49.5	52.4	62.9
Fang et al. [25]	1	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 et al. [26]	1	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
Zhao et al. [4]	1	47.3	60.7	51.4	60.5	61.1	49.9	47.3	68.1	86.2	55.0	67.8	61.0	<b>42.1</b>	60.6	45.3	57.6
Pavlakos et al. [10]	1	48.5	54.4	54.4	52.0	59.4	65.3	49.9	52.9	65.8	71.1	56.6	52.9	60.9	44.7	47.8	56.2
Ci et al. [5]	1	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
Wang et al. [27]	1	44.7	48.9	47.0	49.0	56.4	67.7	48.7	47.0	63.0	78.1	51.1	50.1	54.5	40.1	43.0	52.6
Lee et al. [17]	3	40.2	49.2	47.8	52.6	50.1	75.0	50.2	43.0	55.8	73.9	54.1	55.6	43.3	58.2	43.3	52.8
Marcad et al. [2]	5	44.2	46.7	52.3	49.3	59.9	59.4	47.5	46.2	59.9	65.6	55.8	50.4	52.3	43.5	45.1	51.9
Cai et al. [6]	7	44.6	47.4	45.6	48.8	50.8	59.0	47.2	43.9	57.9	61.9	49.7	46.6	51.3	37.1	39.4	48.8
Xu et al. [28]	7	<b>38.2</b>	<b>44.4</b>	42.8	<b>43.7</b>	47.6	60.3	<b>42.0</b>	45.4	<b>53.2</b>	60.8	<b>46.4</b>	43.5	48.5	34.6	38.6	46.3
Liu et al. [29]	27	44.9	46.7	41.9	45.6	47.9	56.1	44.2	45.5	57.1	59.1	46.8	43.5	47.5	<b>32.6</b>	<b>33.1</b>	46.2
Dario et al. [3]	243	45.2	46.7	43.3	45.6	48.1	55.1	44.6	44.3	57.3	65.8	47.1	44.0	49.0	32.8	33.9	46.8
Our DG-Net	4	41.5	46.6	<b>41.0</b>	44.3	<b>47.1</b>	<b>54.1</b>	44.2	<b>42.5</b>	54.9	<b>58.8</b>	46.9	<b>43.1</b>	46.9	<b>32.6</b>	35.6	<b>45.3</b>
Ci et al. [5]†	1	36.3	38.8	29.7	37.8	34.6	42.5	39.8	32.5	<b>36.2</b>	<b>39.5</b>	34.4	38.4	38.2	31.3	34.2	36.3
Cai et al. [6]†	3	32.9	38.7	32.9	37.0	37.3	44.8	38.7	36.1	41.0	45.6	36.8	37.7	37.7	29.5	31.6	37.2
Marcad et al. [2]†	5	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
Pavlo et al. [3]†	243	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	37.1
Liu et al. [30]†	243	32.3	35.2	33.3	35.8	35.9	41.5	33.2	32.7	44.6	50.9	37.0	32.4	37.0	25.2	27.2	34.7
Our DG-Net†	4	<b>28.5</b>	<b>33.5</b>	<b>28.1</b>	<b>28.9</b>	<b>32.6</b>	<b>35.5</b>	<b>33.3</b>	<b>30.0</b>	37.4	39.9	<b>31.4</b>	<b>30.2</b>	<b>29.5</b>	<b>23.9</b>	<b>25.5</b>	<b>31.2</b>

TABLE I: SOTA Comparison on Human3.6M under Protocol 1. The mark † refers to using GT 2D pose as input of methods.  $T$  is the input sequence length. Note that, our 4-frame DG-Net outperforms 243-frame SOTA method [3].

pose feature enhancement. For example, dynamical spatial unit consists of FSG and DSG in Section III-A. In this case, this unit can first capture physical pose relations from human skeleton, and then adjust relations dynamically according to pose variations in the video. Dynamical temporal unit is a similar case. Finally, we apply the good practices of [4] in our DG-Net, e.g., we use FSG to map the input 2D pose of each frame into the 128-dim pose feature at the beginning. In this case, we can effectively exploit spatial and temporal neighbors for DSG/DTG, by encoding joint location information in a flexible high-dimension space. Additionally, for each type of graph convolution in a dynamical unit, we use the residual-style module built by two layers with 128 feature channels. At the end of a DG-Conv block, we map the final pose feature into the output 3D pose. As suggested in [4], we also add a spatial non-local block after each dynamical unit to enhance holistic pose relations. We repeat our DG-Conv block five times to increase network capacity.

**Multi-Level 3D Pose Supervision.** We propose to supervise DG-Net in a multi-level manner. First, we use 3D pose supervision in the *block level*, i.e., for each DG-Conv block, we supervise the predicted 3D pose by ground truth,

$$\mathcal{L}_{3d}^{block} = \sum_{t=1}^T \sum_{i=1}^N (\hat{\rho}_t^{block,i} - \rho_t^i)^2, \quad (15)$$

where  $\hat{\rho}_t^{block,i} \in \mathbb{R}^3$  is the predicted 3D position of joint  $i$  at frame  $t$ , and  $\rho_t^i \in \mathbb{R}^3$  is the corresponding ground truth. In this case, each block is regularized to be discriminative for predicting effective 3D pose. Second, we use 3D pose supervision in the *network level*, i.e., we concatenate the output features of all DG-Conv blocks together. Subsequently, we map the concatenated feature into the predicted 3D pose, and use ground truth to supervise it,

$$\mathcal{L}_{3d}^{network} = \sum_{t=1}^T \sum_{i=1}^N (\hat{\rho}_t^{network,i} - \rho_t^i)^2. \quad (16)$$

In this case, DG-Net can fuse semantic representations of all DG-Conv blocks together to boost 3D pose estimation. Finally,

we train our DG-Net with both losses,

$$\mathcal{L}_{3d}^{total} = \mathcal{L}_{3d}^{network} + \lambda \sum_{block=1}^{block=5} \mathcal{L}_{3d}^{block}, \quad (17)$$

where  $\lambda$  is a weight coefficient. This allows to train our DG-Net effectively, by leveraging the cooperation power of both block and network levels of 3D pose supervision. In the testing stage, one can simply obtain the predicted 3D pose from the network-level output.

## IV. EXPERIMENTS

### A. Datasets and Implementation Details

We perform DG-Net on widely-used benchmarks in 3D pose estimation, including Human3.6M [34], HumanEva-I [35], and MPI-INF-3DHP [7].

**Human3.6M.** Human3.6M consists of 3.6 million images from 4 different cameras where 7 actors perform 15 activities in the indoor environment. Following the previous works [3], [5], [6], we use subjects 1, 5, 6, 7, 8 for training and subjects 9, 11 for testing. The evaluation metric is the mean per joint position error (MPJPE) between ground truth and estimated 3D pose, which refers to Protocol1. We also report the results of Protocol2, where estimated 3D pose is aligned to ground truth via a rigid transformation. Additionally, following [3], we extract the input 2D pose of Human3.6M from cascaded pyramid network [15].

**HumanEva-I.** HumanEva-I contains video sequences of 3 subjects with 6 actions which are recorded by 3 cameras. Following [3], [11], we train DG-Net and test Protocol2 results for each action. Moreover, as suggested in [3], we extract the input 2D pose of HumanEva-I from Mask R-CNN keypoints detection [36].

**MPI-INF-3DHP.** MPI-INF-3DHP is captured with a multi-view setup, where subjects do not wear visible markers. Following [33], the evaluation metrics of MPI-INF-3DHP are MPJPE, PCK (percentage of correct keypoints), and AUC (area under curve).

Protocol2	$T$	Direct.	Discuss	Eat	Greet	Phone	Photo	Pose	Purch.	Sitting	SitD.	Smoke	Wait	WalkD	Walk.	WalkT.	Avg
Pavlakos et al. [8]	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	51.9
Martinez et al. [11]	1	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	59.4	49.2	45.0	49.5	38.0	43.1	47.7
Fang et al. [25]	1	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
Pavlakos et al. [10]	1	34.7	39.8	41.8	38.6	42.5	47.5	38.0	36.6	50.7	56.8	42.6	39.6	43.9	32.1	36.5	41.8
Ci et al. [5]	1	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
Wang et al. [27]	1	32.8	36.8	42.5	38.5	42.4	49.0	35.4	34.3	53.6	66.2	46.5	34.1	42.3	30.0	39.7	42.2
Lee et al. [17]	3	34.9	<b>35.2</b>	43.2	42.6	46.2	55.0	37.6	38.8	50.9	67.3	48.9	35.2	<b>31.0</b>	50.7	34.6	43.4
Marcard et al. [2]	5	36.9	37.9	42.8	40.3	46.8	46.7	37.7	36.5	48.9	52.6	45.6	39.6	43.5	35.2	38.5	42.0
Cai et al. [6]	7	35.7	37.8	36.9	40.7	39.6	45.2	37.4	34.5	46.9	50.1	40.5	36.1	41.0	29.6	33.2	39.0
Xu et al. [28]	7	<b>31.7</b>	35.3	35.0	35.3	36.9	44.2	32.0	33.8	<b>42.5</b>	49.3	37.6	33.4	39.6	27.6	32.5	36.7
Liu et al. [29]	27	34.0	36.6	33.5	37.2	36.4	42.7	34.1	34.4	45.5	47.0	<b>37.2</b>	33.0	36.6	<b>24.9</b>	<b>26.9</b>	36.0
Pavlo et al. et al. [3]	243	34.1	36.1	34.4	37.2	36.4	42.2	34.4	33.6	45.0	52.5	37.4	33.8	37.8	25.6	27.3	36.5
Our DG-Net	4	32.9	36.1	<b>32.8</b>	<b>35.1</b>	35.8	42.1	<b>31.9</b>	<b>32.2</b>	43.9	<b>46.4</b>	38.0	<b>32.9</b>	35.7	25.5	29.6	<b>35.4</b>
Martinez et al. [11] †	1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	37.1
Ci et al. [5]†	1	24.6	28.6	24.0	27.9	27.1	31.0	28.0	25.0	31.2	35.1	27.6	28.0	29.1	24.3	26.9	27.9
Our DG-Net†	4	<b>20.5</b>	<b>26.0</b>	<b>21.8</b>	<b>23.0</b>	<b>23.7</b>	<b>26.7</b>	<b>25.2</b>	<b>23.1</b>	<b>29.5</b>	<b>32.5</b>	<b>24.2</b>	<b>23.3</b>	<b>23.7</b>	<b>18.7</b>	<b>19.6</b>	<b>24.1</b>

TABLE II: SOTA Comparison on Human3.6M under Protocol2. The mark † refers to using GT 2D pose as input of methods.  $T$  is the input sequence length. Note that, our 4-frame DG-Net outperforms 243-frame SOTA method [3].

Protocol2	$T$	Walking			Jogging			Avg
		S1	S2	S3	S1	S2	S3	
Yasin et al. [32]	1	35.8	32.4	41.6	46.6	41.4	35.4	38.9
Lin et al. [1]	1	26.5	20.7	38.0	41.0	29.7	29.1	30.8
Moreno-Noguer et al. [16]	1	19.7	13.0	<b>24.9</b>	39.7	20.0	21.0	26.9
Pavlakos et al. [8]	1	22.1	21.9	29.0	29.8	23.6	26.0	25.5
Martinez et al. [11]	1	19.7	17.4	46.8	26.9	18.2	18.6	24.6
Fang et al. [25]	1	19.4	16.8	37.4	30.4	17.6	16.3	22.9
Marcard et al. [2]	5	19.1	13.6	43.9	23.2	16.9	15.5	22.0
Pavlo et al. [3]	27	14.5	10.5	47.3	21.9	13.4	13.9	20.2
Our DG-Net	4	<b>13.7</b>	<b>9.5</b>	47.1	<b>21.0</b>	<b>12.6</b>	<b>13.4</b>	<b>19.5</b>

TABLE III: SOTA Comparison on HumanEva-I.

2D-to-3D Pose Lifters	$T$	PCK↑	AUC↑	MPJPE↓
Mono et al. [7]	1	75.7	39.3	-
Mehta et al. [33]	1	79.4	41.6	-
Lin et al. [12]	25	83.6	51.4	79.8
Lin et al. [12]	50	82.4	49.6	81.9
Our DG-Net	4	<b>87.5</b>	<b>53.8</b>	<b>76.0</b>

TABLE IV: SOTA Comparison on MPI-INF-3DHP.

**Implementaion Detail.** Unless stated otherwise, we implement DG-Net as follows. The number of neighbors  $K$  in DSG and DTG are respectively 3 and 4. The weighting function in Eq. (5) and (12) refer to a fully-connected layer. DG-Net contains 5 DG-Conv blocks. The number of input frames is 4. In the loss function of Eq. (17),  $\lambda$  is set as 0.1. We implement DG-Net in PyTorch, where we use Adam optimizer with an initial learning rate of 0.001 and the exponential decay. We train our model for 200 epochs using mini-batches of size 64 on a single GTX 1080 GPU.

### B. SOTA Comparison

. As shown in Table I-IV, our DG-Net achieves the SOTA performance. For example, our 4-frame DG-Net outperforms 243-frame SOTA method [3] on Human3.6M. It is the similar case on HumanEva-I. Note that, the high error on S3 (Walk) of HumanEva-I is due to corrupted mocap data, as indicated by [3]. But still, we achieve a better overall performance. Additionally, as suggested in [12], we mainly compare the

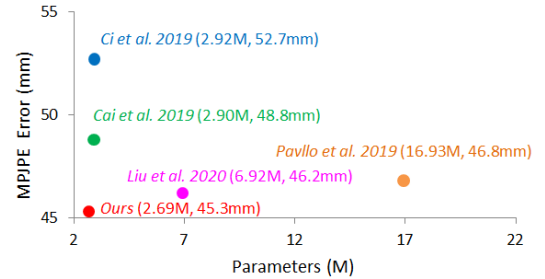


Fig. 4: MPJPE vs. Model Size (Human3.6M). Comparing with [3], our DG-Net achieves a smaller MPJPE but with 84.1% parameter reduction.

$K$ in DSG	1	3	5	7
Our DG-Net	47.3	<b>45.3</b>	48.2	50.8
$K$ in DTG	1	3	5	7
Our DG-Net	48.0	46.5	47.3	49.2

TABLE V: The number of joint neighbors  $K$  used in DSG and DTG.

previous 2D-to-3D pose lifters for MPI-INF-3DHP, in order to show generalization capacity of our model. In this case, the ground truth 2D pose is given as input for all the methods. Clearly, our DG-Net outperforms other lifters on all the evaluation metrics. Finally, our DG-Net achieves the best balance between model size and estimation error, as shown in Fig. 4. All these facts demonstrate the effectiveness of DG-Net on 3D pose estimation in videos.

### C. Ablation Study

Without loss of generality, we investigate ablation studies on Human3.6M.

**The number of joint neighbors  $K$  in DSG and DTG.** When we change  $K$  in DSG (or DTG), we fix  $K$  in DTG (or DSG) as 4 (or 3). As expected, when we increase  $K$  in DSG (or DTG) in Table V, the estimation error of DG-Net first decreases and then increases. The main reason is that, when a joint is connected with too few (or many) neighbors, its pose

Graph \ Weighting	W/O Weighting	FC	EG
Random	83.4	80.5	80.9
Fixed	52.7	50.3	51.4
Full	58.8	56.3	57.3
Symmetry	53.2	53.0	53.0
Precomputed	50.4	48.1	48.8
Dynamical	48.1	45.3	46.9

TABLE VI: Graph Connection Styles & Weighting Functions in DSG and DTG. FC means Fully Connected; EG means Embedded Gaussian

Dynamical Spatial Unit	MPJPE
<i>FSG</i>	57.9
<b><i>FSG+DSG</i></b>	<b>55.8</b>
<i>FSG+FSG</i>	57.4
<i>FSG+DSG+DSG</i>	57.4
<i>FSG+DSG+FSG</i>	56.5
Dynamical Temporal Unit	MPJPE
<b><i>FSG+DSG+FTG</i></b>	52.0
<b><i>FSG+DSG+FTG+DTG</i></b>	47.0
<b><i>FSG+DSG+FTG+FTG</i></b>	51.3
<b><i>FSG+DSG+FTG+DTG+DTG</i></b>	47.6
<b><i>FSG+DSG+FTG+DTG+FTG+FTG</i></b>	<b>45.3</b>
<b><i>FSG+DSG+FTG+DTG+FTG+DTG</i></b>	46.1
<b><i>FSG+DSG+FTG+DTG+FTG+FTG</i></b>	45.5

TABLE VII: DG-Conv Block.

context tends to be insufficient (or noisy). Hence, we choose the moderate  $K$  (3/4 for DSG/DTG) in our experiments.

**Graph Connection Styles & Weighting Functions in DSG and DTG.** For graph connection style, we investigate six settings. The random setting is to randomly select  $K$  neighbors for each joint. The full setting is to connect each joint with all other joints. The fixed setting is to connect joints according to human skeleton. The symmetry setting refers to the symmetrical connection of joints in [25]. The precomputed setting is to find  $K$  neighbors for each joint by the estimated 2D pose, and then fix these connections in DSG and DTG. The dynamical setting is our design. For the weighting function, we investigate three settings. Except the without setting, we use a FC layer or the kernel in [37] to obtain the importance score. We denote them as the fully connected or embedded gaussian settings. As shown in Table VI, the dynamical setting achieves the best among all the graph styles, showing the effectiveness of our design. Moreover, the fully connected setting of weighting mechanism is preferable. Hence, we choose this setting in our experiment.

**Dynamical Spatial/Temporal Units.** We gradually introduce our DSG and DTG operations in DG-Conv Block. As shown in Table VII, the performance achieves the best with (dynamical spatial unit: *FSG+DSG*)+(dynamical temporal unit: *FTG+DTG+FTG*). Hence, we choose this setting in our experiment.

**Blocks & Frames.** In Table VIII, the performance is the best with 5 DG-Conv blocks. Additionally, the 4-frame setting is comparable with the 5-frame setting. To reduce computation cost, we choose the 4-frame setting.

**Other Detailed Designs.** We further investigate other de-

Blocks	1	3	5	Frames(T)	2	3	4	5
MPJPE	60.7	47.0	<b>45.3</b>	MPJPE	51.2	48.1	<b>45.3</b>	<b>45.3</b>

TABLE VIII: No. of DG-Conv Blocks & Input Frames.

Supervision	Network Loss	Total Loss
Our DG-Net	48.7	<b>45.3</b>
Nonlocal	Without	With
Our DG-Net	50.1	<b>45.3</b>
Coordinate Operation	Non-Balanced	Balanced
Our DG-Net	46.1	<b>45.3</b>
Choice of 2D Estimator	SH [14]	CPN [15]
Pavlo et al. [3]	53.4	46.8
Our DG-Net	<b>49.9</b>	<b>45.3</b>

TABLE IX: Detailed Designs (Supervision, Nonlocal, Coordinate Operation, Choice of 2D Pose Estimator).

tailed designs in Table IX. First, we evaluate our multi-level supervision. As expected, the total loss in Eq. (17) leads to a better performance, compared to the network loss in Eq. (16). It illustrates that, adding 3D supervision in each block can effectively regularize our DG-Net with more discriminative pose representations. Second, we follow the good practice in [4] via embedding spatial nonlocal operations in our DG-Conv block. As expected, it achieves a better performance by learning global pose relations. Third, we evaluate the impact of coordinate operations. In general, there are two coordinate systems for 3D human pose. The first uses root-relative 3D joint coordinates in the camera coordinate system. The second refers to concatenation of the predicted depth of each joint and UV coordinates extracted from 2D pose detector. We simply average 3D pose estimation from two coordinate systems for balancing. As expected, such balance takes advantage of both systems to achieve a better 3D estimation. Finally, we evaluate the impact of 2D pose estimators. With a more accurate 2D pose estimator [15], the performance is better.

#### D. Cross Actions Generalization Ability

We train our model on one of the 15 actions in the Human3.6M [34] dataset and test on all actions. As shown in Figure 5, our DG-Net significantly improves the performance compared with baseline and outperforms another SOTA video 2D-pose lifter [6] by a large margin. This demonstrates that our DG-Net is capable of adjusting human-joint affinity dynamically to enhance 3D pose estimation, depending on the personalized poses of different actions in the video.

	Moreno-Nouguer [16]	Martinez [11]	Hossain [2]	Our
GT/GT	62.17	37.10	31.6	<b>24.17</b>
GT/GT + $\mathcal{N}(0, 5)$	67.11	46.65	37.46	<b>31.86</b>
GT/GT + $\mathcal{N}(0, 10)$	79.12	52.84	49.41	<b>38.63</b>
GT/GT + $\mathcal{N}(0, 15)$	96.08	59.97	61.80	<b>46.92</b>
GT/GT + $\mathcal{N}(0, 20)$	115.55	70.24	73.65	<b>56.33</b>

TABLE X: Robustness to noise. The results of PA-MPJPE under protocol2 are reported.

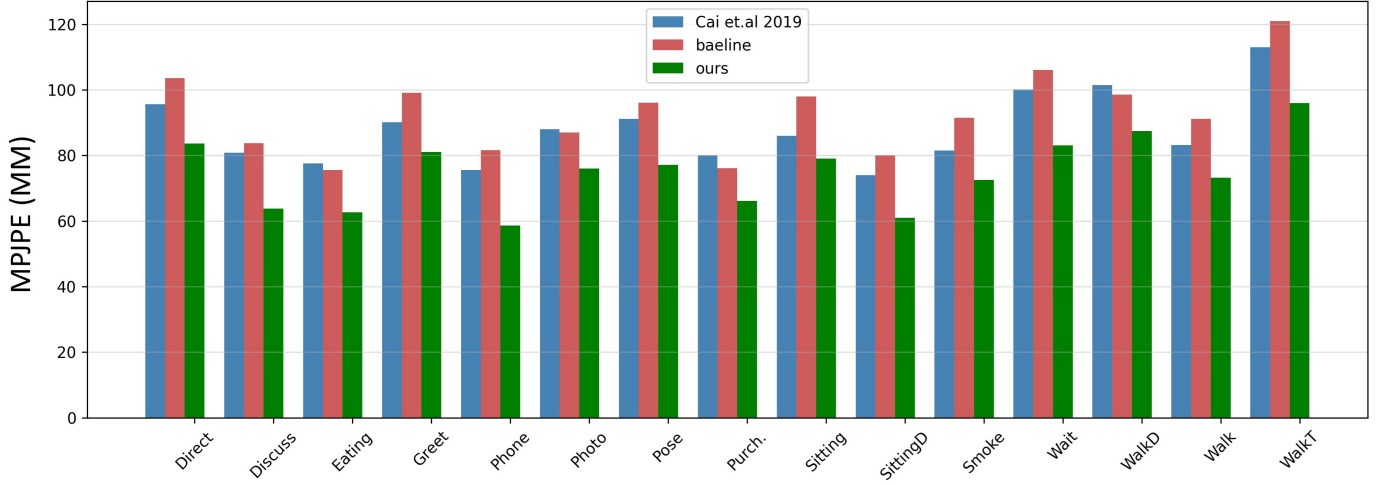


Fig. 5: The 3D pose estimation errors when we train the video-level 2D pose lifter [6], baseline and our DG-Net on each of the 15 actions and test on all actions. X-axis indicates the action used in training.

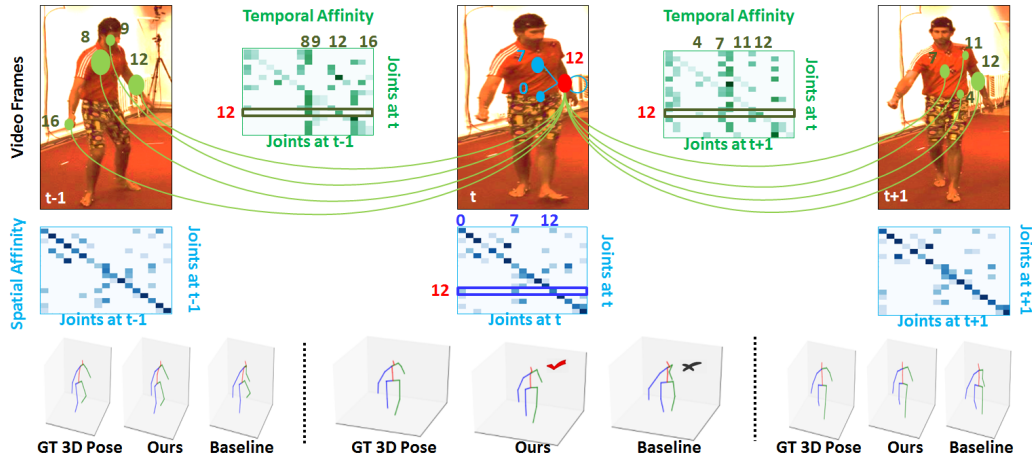


Fig. 6: Visualization. We show the estimated 3D pose in a *WalkingDog* video. As expected, spatial and temporal affinity matrices are dynamically adjusted, which allows DG-Net to reduce ambiguity by learning richer pose relations.

#### E. Robustness to Noise

We evaluate the robustness of our approach ( $T=4$ ) to corrupted 2D joint locations. In this experiment, all approaches are trained using the ground truth 2D and 3D pairs in Human3.6M dataset. Then we test on inputs corrupted by different levels of Gaussian noise. As shown in Table X, the first four lines are trained on ground truth 2D pose, and then tested with different degrees of Gaussian noise added on ground truth 2D pose. All results are tested under protocol2 on Human3.6M dataset. Our method obtains the best result and maintains robustness to all levels of Gaussian noise as well as 2D pose detector. Our specific designed important scores mechanism in DG-Net and holistic description of the whole body in one frame provided by spatial nonlocal play significant roles in robustness to noise.

#### F. Visualization

We visualize DG-Net via the estimated 3D pose in a *WalkingDog* video. Moreover, we show spatial affinity (per frame) in DSG, and temporal affinity (between two frames)

in DTG. All these affinity matrices are extracted from the 1st DG-Conv block. As shown in Fig. 6, spatial/temporal human-joint affinity matrices change over time. This allows DG-Net to dynamically exploit pose context for each joint to boost 3D estimation, e.g., for Joint12 (left ankle) at  $t$ , spatial affinity discovers Joint0 (hip) and Joint7 (spine) as extra spatial context at  $t$ , while temporal affinity discovers Joint4 (left hip), Joint7 (spine), Joint11 (left shoulder) as extra temporal context at  $t+1$ , and Joint8 (thorax), Joint9 (nose), Joint16 (right wrist) as extra temporal context at  $t-1$ . Those joints can effectively tell that, Joint12 is straighten and moving with the upper body. As a result, our DG-Net correctly estimate Joint12 at  $t$ . On the contrary, baseline distorts this joint with a wrong depth, due to the fixed spatial and temporal graph convolutions. Additional qualitative results on Human3.6M [34], MPI-INF-3DHP [7] and Upenn Action [38] are shown in Figure 7. Our model also achieves good results in outdoor scenes and in the wild, which proves the generalization of our model.



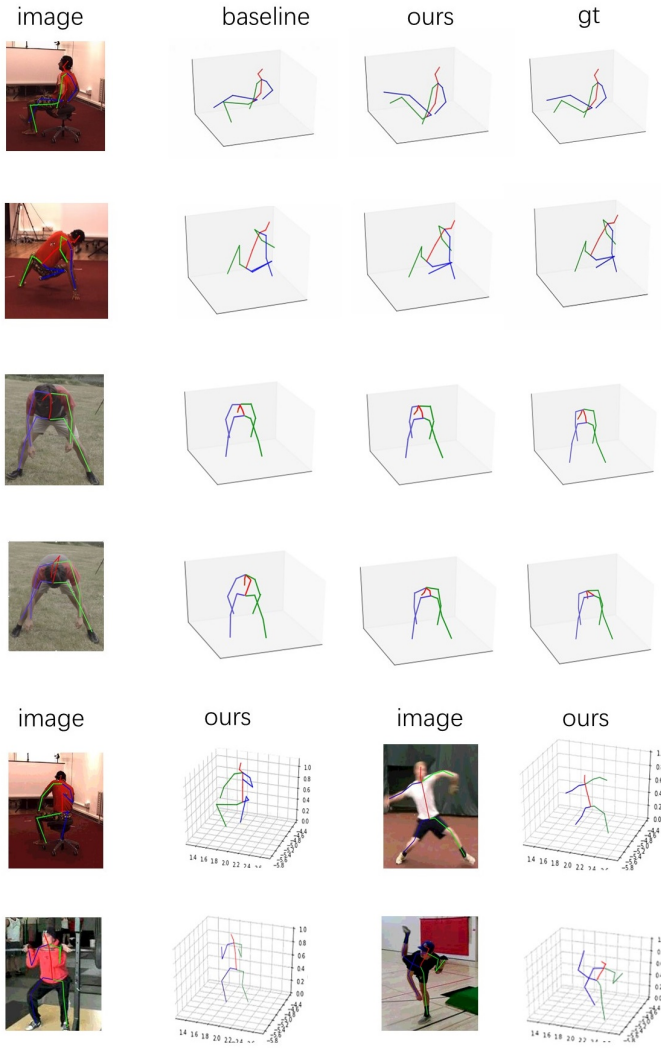


Fig. 7: Visualization.

## V. CONCLUSION

In this work, we propose a novel DG-Net for video 3D pose estimation. By learning human-joint affinity dynamically, it can build up effective joint relations to reduce spatial and temporal ambiguity caused by complex pose variations in videos. Extensive experiments demonstrate high accuracy and effectiveness of our DG-Net.

## REFERENCES

- [1] M. Lin, L. Lin, X. Liang, K. Wang, and H. Cheng, "Recurrent 3d pose sequence machines," in *CVPR*, July 2017. **1, 6**
- [2] M. R. I. Hossain and J. Little, "Exploiting temporal information for 3d human pose estimation," *ECCV*, 2018. **1, 2, 5, 6, 7**
- [3] D. Pavlo, C. Feichtenhofer, D. Grangier, and M. Auli, "3d human pose estimation in video with temporal convolutions and semi-supervised training," in *CVPR*, 2019. **1, 2, 5, 6, 7**
- [4] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. N. Metaxas, "Semantic graph convolutional networks for 3d human pose regression," *CVPR*, 2019. **1, 2, 4, 5, 7**
- [5] H. Ci, C. Wang, X. Ma, and Y. Wang, "Optimizing network structure for 3d human pose estimation," in *ICCV*, October 2019. **1, 2, 5, 6**
- [6] Y. Cai, L. Ge, J. Liu, J. Cai, T.-J. Cham, J. Yuan, and N. M. Thalmann, "Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks," in *ICCV*, October 2019. **1, 2, 5, 6, 7, 8**

- [7] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation in the wild using improved cnn supervision," in *3DV*, 2017. **2, 5, 6, 8**
- [8] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *CVPR*, 2017. **2, 6**
- [9] X. Sun, B. Xiao, F. Wei, S. Liang, and Y. Wei, "Integral human pose regression," in *ECCV*, 2018. **2**
- [10] G. Pavlakos, X. Zhou, and K. Daniilidis, "Ordinal depth supervision for 3d human pose estimation," in *CVPR*, 2018. **2, 5, 6**
- [11] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3d human pose estimation," in *ICCV*, 2017. **2, 4, 5, 6, 7**
- [12] J. Lin and G. H. Lee, "Trajectory space factorization for deep video-based 3d human pose estimation," *arXiv preprint arXiv:1908.08289*, 2019. **2, 6**
- [13] J. Wang, S. Yan, Y. Xiong, and D. Lin, "Motion guided 3d pose estimation from videos," *arXiv preprint arXiv:2004.13985*, 2020. **2**
- [14] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," 2016. **2, 7**
- [15] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *CVPR*, 2018. **2, 5, 7**
- [16] F. Moreno-Noguer, "3d human pose estimation from a single image via distance matrix regression," in *CVPR*, 2017. **2, 6, 7**
- [17] K. Lee, I. Lee, and S. Lee, "Propagating lstm: 3d pose estimation based on joint interdependency," in *ECCV*, September 2018. **2, 5, 6**
- [18] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," in *arXiv*, 2015. **2**
- [19] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," in *ICLR*, 2016. **2**
- [20] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, 2017. **2**
- [21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *ICLR*, 2017. **2**
- [22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018. **2**
- [23] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *AAAI*, 2018. **2, 4**
- [24] Y. Chen, L. Zhao, X. Peng, J. Yuan, and D. N. Metaxas, "Construct dynamic graphs for hand gesture recognition via spatial-temporal attention," *BMVC*, 2019. **2**
- [25] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3d pose estimation," in *AAAI*, 2018. **5, 6, 7**
- [26] W. Yang, W. Ouyang, X. Wang, J. Ren, H. Li, and X. Wang, "3d human pose estimation in the wild by adversarial learning," in *CVPR*, 2018. **5**
- [27] J. Wang, S. Huang, X. Wang, and D. Tao, "Not all parts are created equal: 3d pose estimation by modeling bi-directional dependencies of body parts," in *ICCV*, October 2019. **5, 6**
- [28] J. Xu, Z. Yu, B. Ni, J. Yang, X. Yang, and W. Zhang, "Deep kinematics analysis for monocular 3d human pose estimation," in *CVPR*, June 2020. **5, 6**
- [29] J. Liu, Y. Guang, and J. Rojas, "Gast-net: Graph attention spatio-temporal convolutional networks for 3d human pose estimation in video," *arXiv preprint arXiv:2003.14179*, 2020. **5, 6**
- [30] R. Liu, J. Shen, H. Wang, C. Chen, S.-c. Cheung, and V. Asari, "Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction," in *CVPR*, June 2020. **5**
- [31] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *CVPR*, 2018, pp. 6450–6459. **4**
- [32] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall, "A dual-source approach for 3d pose estimation from a single image," in *CVPR*, June 2016. **6**
- [33] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt, "Vnect: Real-time 3d human pose estimation with a single rgb camera," vol. 36, no. 4, 2017. [Online]. Available: <http://gsv.mpi-inf.mpg.de/projects/VNect/> **5, 6**
- [34] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *PAMI*, 2014. **5, 7, 8**
- [35] L. Sigal, A. O. Balan, and M. J. Black, "Human3.6m: Synchronized video and motion capture dataset and baseline algorithm forevaluation of articulated human motion," in *IJCV*, 2010. **5**

- [36] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” *ICCV*, 2017. 5
- [37] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” *CVPR*, 2018. 7
- [38] W. Zhang, M. Zhu, and K. G. Derpanis, “From actemes to action: A strongly-supervised representation for detailed action understanding,” in *ICCV*, 2013, pp. 2248–2255. 8