Modulated Graph Convolutional Network for 3D Human Pose Estimation

Zhiming Zou and Wei Tang*
University of Illinois at Chicago
{zzou6, tangw}@uic.edu

Abstract

The graph convolutional network (GCN) has recently achieved promising performance of 3D human pose estimation (HPE) by modeling the relationship among body parts. However, most prior GCN approaches suffer from two main drawbacks. First, they share a feature transformation for each node within a graph convolution layer. This prevents them from learning different relations between different body joints. Second, the graph is usually defined according to the human skeleton and is suboptimal because human activities often exhibit motion patterns beyond the natural connections of body joints. To address these limitations, we introduce a novel Modulated GCN for 3D HPE. It consists of two main components: weight modulation and affinity modulation. Weight modulation learns different modulation vectors for different nodes so that the feature transformations of different nodes are disentangled while retaining a small model size. Affinity modulation adjusts the graph structure in a GCN so that it can model additional edges beyond the human skeleton. We investigate several affinity modulation methods as well as the impact of regularizations. Rigorous ablation study indicates both types of modulation improve performance with negligible overhead. Compared with state-of-the-art GCNs for 3D HPE, our approach either significantly reduces the estimation errors, e.g., by around 10%, while retaining a small model size or drastically reduces the model size, e.g., from 4.22M to 0.29M (a 14.5× reduction), while achieving comparable performance. Results on two benchmarks show our Modulated GCN outperforms some recent states of the art. Our code is available at `https://github.com/ZhimingZo/Modulated-GCN`.

1. Introduction

3D human pose estimation (HPE) aims to accurately recover the 3D locations of body joints in the camera coordinate system from a single image. It plays an important role in several valuable applications such as human-computer interaction, action recognition, and intelligent surveillance. However, 3D HPE remains a challenging problem due to its ill-posed nature. Multiple valid 3D body configurations can be projected to the same 2D pose in the image space.

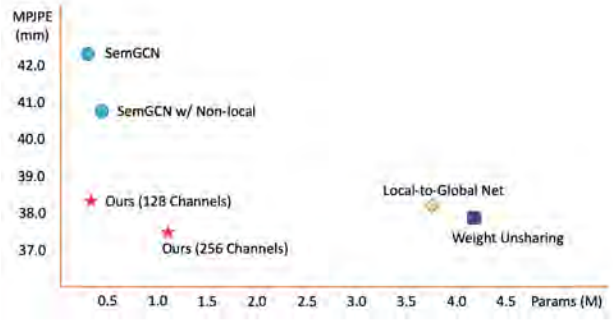

Figure 1. Comparison of the performance and model size between the proposed Modulated GCN and state-of-the-art GCNs designed for 3D HPE, *i.e.*, SemGCN [57], Local-to-Global Net [3], and Weight Unsharing [28]. A lower MPJPE value indicates better performance. All methods are evaluated on Human3.6M [15] with ground truth 2D joints as input.

State-of-the-art 3D HPE systems are built on deep neural networks [20] due to their strong capability to learn effective feature representations from data. Some approaches [61, 35, 44, 37, 43, 54, 23, 4] regress 3D joint coordinates or heat maps directly from a monocular image via a convolutional neural network (CNN) [21, 19]. Recent works decompose the problem into two subtasks, *i.e.*, 2D HPE followed by 2D-to-3D pose lifting [31, 3, 6, 57, 38, 30, 50, 52, 5, 63, 29, 48]. For example, Martinez *et al*. [31] construct a simple fully connected network taking only 2D keypoints as input and yield promising 3D HPE performance.

Recently, graph convolutional networks (GCNs) have been applied for 3D HPE [57, 3, 6, 28] to model the correlation between body joints. They repeatedly transform and aggregate features of neighboring nodes to obtain more powerful feature representations. Their superior performance over the fully connected networks proves that relational reasoning is critical to mitigate the depth ambiguity.

However, most previous GCNs suffer from two limita-

*Corresponding author.

tions. First, they share a feature transformation for each node within a graph convolution layer. Since it is the feature transformation that captures the relations between each node and their neighboring nodes, this kind of weight sharing prevents the GCN from learning diverse relational patterns between different body joints. Recently, Liu *et al*. [28] solve this problem via *weight unsharing* and apply different feature transformations to different nodes before aggregating their features. However, it significantly increases the model size, *i.e*., by a factor of the number of body joints (typically 16 or 17). Second, the graph in a GCN defines the pairwise correlations between body joints, and it is usually defined according to the human skeleton. However, human activities often exhibit motion patterns beyond the natural connections of body joints, *e.g*., the strong correlation between arms and legs for a walking or running person. It remains unclear what kind of graph structure is optimal for 2D-to-3D pose lifting.

This paper introduces a novel approach, termed the Modulated GCN, to resolve these two difficult issues. It consists of two main components: weight modulation and affinity modulation. Unlike weight unsharing [28] which applies different weight matrices to different nodes, weight modulation uses a shared weight matrix as in the vanilla GCN but learns different modulation vectors for different nodes. By manipulating the latent weight space, the feature transformations of different nodes are disentangled. This enables the graph convolution to learn diverse relationships between different body joints while *retaining a small model size*. Affinity modulation means to adjust the graph structure in a GCN so that each graph convolution layer focuses on additional edges beyond the human skeleton. This is achieved by learning a modulation matrix added to the human skeleton affinity matrix. However, the unconstrained modulation can be suboptimal as the correlation patterns of body joints exhibit certain properties. This motivates us to study what kind of prior should be enforced on the affinity modulation. Specifically, we have an in-depth investigation on whether symmetry, sparsity and low-rank constraints help improve the generalization ability.

In sum, the contribution of this paper is threefold.

- We introduce weight modulation to disentangle the feature transformations of different nodes. It enables the GCN to learn diverse relational patterns between different body joints while *maintaining a small model size*.
- We investigate different affinity modulation methods as well as the impact of different regularizations. Our optimal affinity modulation helps each graph convolution layer focus on additional edges beyond the skeleton graph.
- Compared with state-of-the-art GCN methods, our Modulated GCN addresses the dilemma between the accuracy and model complexity, as shown in Fig. 1. It significantly reduces the model size of the latest Weight Unsharing GCN [28], *i.e*., from 4.22M to 0.29M (a 14.5$\times$ reduction), while achieving similar accuracy. It reduces the estimation error of Semantic GCN [57] by around 10% (9.2% on MPJPE and 10.3% on P-MPJPE) while maintaining a small model size.

2. Related Work

2.1. 3D Human Pose Estimation

Lee and Chen [22] first investigate the problem of inferring the 3D body configuration from a single image. Later approaches [16, 13] use the estimated 2D pose as a query to retrieve the nearest 3D pose from a large pose library. Recent state-of-the-art approaches are based on deep neural networks. They can be broadly divided into two categories.

The first category is to train deep convolutional neural networks (CNNs) to directly regress 3D human poses from input images in an end-to-end fashion [44, 4, 37, 33, 61, 35, 23, 45, 60]. Some of them strive to learn a more powerful representation. For example, Park *et al*. [35] concatenate 2D pose estimation as well as information on relative positions with respect to multiple joints to obtain a more accurate 3D pose. Pavlakos *et al*. [37] introduce a fine discretization of the 3D space around the subject and train a CNN to predict per voxel likelihoods for each joint. Chen *et al*. [4] build a part-aware 3D pose estimator by searching a set of network architectures to learn heterogeneity among human body parts. Some other works attempt to integrate 3D geometry modeling into deep learning. Zhou *et al*. [62] leverage a sparsity-driven 3D geometric prior and temporal smoothness to infer 3D poses from uncertain 2D keypoint maps via the EM algorithm. Zhou *et al*. [61] directly embed a kinematic object model into the deep neutral network learning for general articulated object pose estimation. The approaches in the first category benefit from the rich information contained in images. However, it cannot be generalized well to different environments, *e.g*., from indoor to outdoor environment.

The second category of approaches [31, 3, 6, 57, 50, 52, 28, 30, 5, 55] decouple 3D HPE into well-studied 2D pose regression and 2D-to-3D pose lifting from the detected 2D keypoints. Some approaches [28, 57, 3, 6] exploit GCN for 3D HPE, which are most related to our work. Cai *et al*. [3] construct a local-to-global network which enables GCN to learn multi-scale feature representations corresponding to different semantic meanings. Ci *et al*. [6] enhance the representation capability of GCN by introducing a locally connected network.

Zhao *et al*. [57] propose a semantic GCN by multiplying a learnable mask to the skeleton-based affinity matrix and

then extending it to channel-specific affinity masks. Our Modulated GCN differs from theirs in two aspects. First, a shared feature transformation is employed by all nodes in the semantic GCN while we explicitly disentangle the feature transformations for different body joints by learning node-specific modulation vectors. Second, we find multiplying a learnable mask to the skeleton-based affinity matrix is suboptimal because it cannot learn relations beyond the natural connections of body joints. We have an in-depth investigation of different affinity modulation methods to resolve this issue. In addition, we have a comprehensive study of the impact of different regularizations on the affinity modulation.

Liu *et al*. [28] investigate different weight unsharing methods in a GCN. Our work is different from theirs in two aspects. First, weight unsharing uses different transformation matrices for different nodes and thus significantly increases the model size. By contrast, our weight modulation solves this problem by introducing a modulation vector for each node. Second, we have an in-depth investigation of affinity modulation, but they ignore.

2.2. Graph Convolutional Networks (GCNs)

GCNs [18, 7, 11, 2] generalize the capability of CNNs by performing convolution operations on graph-structured data. Existing GCNs can be divided into two main streams, the spectral-based approaches [7, 24, 42] and the spatial-based approaches [18, 11, 2, 46]. The first stream are defined in the Fourier domain by computing eigen-decomposition of the graph Laplacian [1, 8]. Our work falls into the second stream which define the convolutional filters in the vertex domain and directly apply convolution operations on the graph nodes and their neighbors [56].

GCNs have been applied in other computer vision tasks besides 3D HPE, such as action recognition [58, 41], object detection [51], visual question answering [26], and object tracking [10]. Shi *et al*. [41] add a learnable matrix to the affinity matrix in a GCN for action recognition. Our approach differs with theirs in several aspects. First, we have an in-depth investigation of different affinity modulation methods and draw new conclusions they do not have. Second, we study the impact of several regularizations of the modulation matrix on the generalization ability, but they ignore. Third, we also introduce the new weight modulation. Fourth, we focus on 3D HPE while they focus on action recognition.

3. Our Approach

Our Modulated GCN contains two major components: weight modulation and affinity modulation. We first review the vanilla GCN and its weight unsharing variant [28] in Sec. 3.1. Then, weight modulation and affinity modulation are introduced in Sec. 3.2 and Sec. 3.3, respectively. Finally, we present the network architecture in Sec. 3.4.

3.1. Vanilla GCN

We briefly review the vanilla GCN introduced by [18]. A graph is defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a set of N nodes and $\mathcal{E}$ is a collection of edges. The edges can be represented by an affinity matrix $\mathbf{A} \in \{0, 1\}^{N \times N}$. Each node i is associated with a D-dimensional feature vector $\mathbf{h}_i \in \mathbb{R}^D$. The collection of features of all nodes can be written as a matrix $\mathbf{H} \in \mathbb{R}^{D \times N}$, where the ith column of $\mathbf{H}$ is $\mathbf{h}_i$. A graph convolutional layer transforms and aggregates the input features following the equation below:

$$\mathbf{H}' = \sigma(\mathbf{W}\mathbf{H}\tilde{\mathbf{A}}) \tag{1}$$

where $\mathbf{H}' \in \mathbb{R}^{D' \times N}$ is the updated feature matrix, $\sigma(\cdot)$ is the activation function, *i.e*., ReLU [34], and $\mathbf{W} \in \mathbb{R}^{D' \times D}$ is the learnable weight matrix which changes the feature dimension from D to D'. $\tilde{\mathbf{A}}$ is the symmetrically normalized affinity matrix [18]. A GCN obtains enhanced feature representations by stacking multiple graph convolutional layers and repeatedly transforming and aggregating the features of nodes and their neighbors. The enhanced feature representations are used for prediction via the last layer in the network.

Let $\tilde{a}_{ij}$ be the (i, j)th entry of $\tilde{\mathbf{A}}$, $\mathcal{N}_i$ and $\tilde{\mathcal{N}}_i \equiv \mathcal{N}_i \cup \{i\}$ represent the set of neighboring nodes of node i excluding and including itself respectively. Note that $j \in \tilde{\mathcal{N}}_i$ if and only if $\tilde{a}_{ij}$ is non-zero. Eq. (1) can be equivalently written as:

$$\mathbf{h}'_i = \sigma(\sum_{j \in \tilde{\mathcal{N}}_i} \mathbf{W}\mathbf{h}_j \tilde{a}_{ij}) \tag{2}$$

where $\mathbf{h}'_i$ is the ith column of the updated feature matrix $\mathbf{H}'$, $i \in \{1, ..., N\}$. One limitation of this vanilla graph convolution is that it shares a feature transformation $\mathbf{W}$ for each node and thus prevents the GCN from learning diverse relational patterns between different body joints. Liu *et al*. [28] solve this problem by using a different weight matrix $\mathbf{W}_j \in \mathbb{R}^{D' \times D}$ to transform each node j before aggregating them:

$$\mathbf{h}'_i = \sigma(\sum_{j \in \tilde{\mathcal{N}}} \mathbf{W}_j \mathbf{h}_j \tilde{a}_{ij}) \tag{3}$$

They also found decoupling the transformations of self-connections and other edges can significantly improve the 3D HPE performance, which has also been observed in other works [28, 57, 53]. While weight unsharing using Eq. (3) improves the performance, it significantly increases the model size, *i.e*., by a factor of N (typically 16 or 17).

3.2. Weight Modulation

Weight modulation means to solve the aforementioned problem caused by a shared feature transformation for different nodes while retaining a small model size. Unlike

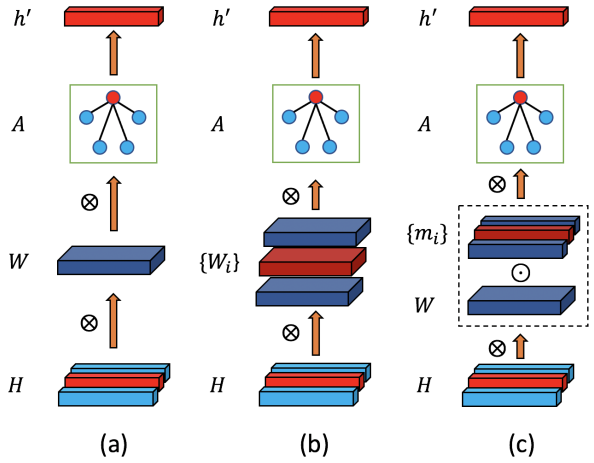

Figure 2. Illustration of (a) the vanilla graph convolution, (b) weight unsharing, and (c) weight modulation. $\otimes$ and $\odot$ denote matrix multiplication and element-wise multiplication, respectively. (a) The vanilla graph convolution uses a shared weight matrix for all nodes. (b) Weight unsharing [28] assigns different weight matrices to different nodes. (c) The proposed weight modulation uses a shared weight matrix but learns different modulation vectors for each node.

weight unsharing in Eq. (3), weight modulation uses a shared weight matrix $\mathbf{W}$ as in the vanilla GCN but learns a different modulation vector $\mathbf{m}_i$ for each node i and uses it to *modulate* the shared weight matrix:

$$\mathbf{h}'_i = \sigma(\sum_{j \in \tilde{\mathcal{N}}_i} (\mathbf{m}_j \odot \mathbf{W})\mathbf{h}_j \tilde{a}_{ij}) \tag{4}$$

where $\mathbf{m}_j \in \mathbb{R}^{D'}$ is a learnable modulation vector for node j ($j = 1, \ldots, N$), $\odot$ denotes element-wise multiplication but should broadcast properly. Specifically, $\mathbf{m}_j \odot \mathbf{W}$ means the dth row of $\mathbf{W}$ is scaled by the dth element of $\mathbf{m}_j$, and the result is of the same dimension as $\mathbf{W}$, *i.e.*, $D' \times D$.

If we treat weight sharing in Eq. (1) and weight unsharing in Eq. (3) as two extremes, weight modulation lies between them. On the one hand, the feature transformations applied to different nodes are different as their modulation vectors are different. On the other hand, these different transformations lie in a common subspace. We will show that weight modulation can solve the problem caused by weight sharing as effectively as weight unsharing. Unlike weight unsharing which significantly increases the model size, the number of additional parameters brought by weight modulation is ignorable. This further makes weight modulation generalize better. Specifically, the numbers of parameters of the vanilla graph convolution, weight unsharing and weight modulation are respectively $D' \times D$, $D' \times D \times N$ and $D' \times (D + N)$. For 3D HPE, N is significantly smaller than D and D'.

Putting together updated features of all nodes, Eq. (4) can be equivalently written as a compact form:

$$\mathbf{H}' = \sigma((\mathbf{M} \odot (\mathbf{W}\mathbf{H}))\tilde{\mathbf{A}}) \tag{5}$$

where $\mathbf{M} \in \mathbb{R}^{D' \times N}$ is the collection of all modulation vectors, and its ith column is $\mathbf{m}_i$. From Eq. (5), we can also understand weight modulation as multiplying different modulation vectors to updated feature vectors of different nodes before they are aggregated.

3.3. Affinity Modulation

The vanilla GCN [18] exploits a predefined affinity matrix to capture the correlations between nodes. For 3D HPE, the graph is usually defined based on the human skeleton. We call it a skeleton graph and denotes it as $\mathbf{A}_{skeleton} \in \mathbb{R}^{N \times N}$. An element in $\mathbf{A}_{skeleton}$ is 1 if the corresponding pair of body joints are naturally connected and 0 otherwise. Recently, Zhao *et al*. [57] find it beneficial to multiply a learnable mask $\mathbf{P} \in \mathbb{R}^{N \times N}$ to $\mathbf{A}_{skeleton}$ so that the nonzero values in it can be adjusted:

$$\mathbf{A}_{mul} = \mathbf{A}_{skeleton} \odot \mathbf{P} \tag{6}$$

where $\mathbf{A}_{mul} \in \mathbb{R}^{N \times N}$ is a new affinity matrix.

One limitation of this method is that only the affinity values corresponding to the edges in the skeleton graph are learnable. Human activities often exhibit motion patterns beyond the natural connections of body joints, *e.g.*, the strong correlation between arms and legs for a walking or running person. A simple solution is to replace the multiplication in Eq. (6) with addition or use their mixed form:

$$\mathbf{A}_{add} = \mathbf{A}_{skeleton} + \mathbf{Q} \tag{7}$$

$$\mathbf{A}_{mix} = \mathbf{A}_{skeleton} \odot \mathbf{P} + \mathbf{Q} \tag{8}$$

where $\mathbf{Q} \in \mathbb{R}^{N \times N}$ is a learnable matrix. We also consider a learnable affinity matrix without imposing any skeleton prior:

$$\mathbf{A}_{no-skeleton} = \mathbf{Q} \tag{9}$$

Note in Eqs. (6)-(9), $\mathbf{P}$ and $\mathbf{Q}$ always represent learnable matrices and $\mathbf{A}_{skeleton}$ is a constant affinity matrix.

$\mathbf{A}_{add}$, $\mathbf{A}_{mix}$ and $\mathbf{A}_{no-skeleton}$ are more flexible than $\mathbf{A}_{mul}$ because they allow the graph to include extra edges beyond the natural connections of body joints. However, too much freedom can lead to overfitting and harm the generalization ability of a GCN. We have an in-depth investigation of this potential issue by exploring the impact of different regularizations on affinity modulation. Specifically, we consider three types of regularizations: symmetry, low rank and sparsity.

Symmetry. A symmetric affinity matrix corresponds to an undirected graph. It means the correlation between two

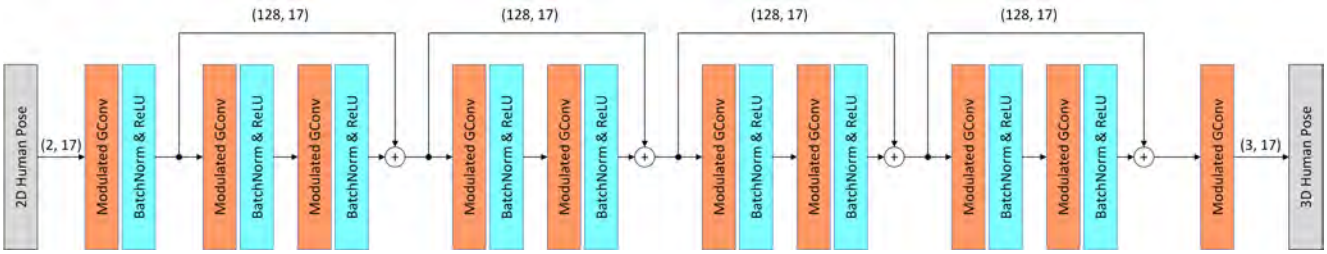

Figure 3. The network architecture of the proposed Modulated GCN for 3D human pose estimation. (D, N) indicates the feature channels and the number of body joints, respectively.

body joints does not depend on the direction of their connection. Since the skeleton graph is symmetric, we only need to make the modulation matrix symmetric to get a symmetric affinity matrix. This can be easily achieved by representing the modulation matrix as the average of a matrix and its transpose: $(\mathbf{Q} + \mathbf{Q}^\top)/2$.

Low rank. An alternative way to enforce the affinity matrix to be symmetric is to represent the modulation matrix as the multiplication between a matrix $\mathbf{L} \in \mathcal{R}^{N \times C}$ and its transpose: $\mathbf{L}\mathbf{L}^\top \in \mathcal{R}^{N \times N}$. If $C < N$, this representation also constrains the modulation matrix to be within a low-dimensional subspace.

Sparsity. As a common regularization technique in machine learning, sparsity promotes generalization [32] by removing irrelevant or weakly relevant features. A sparse affinity matrix has only a few nonzero elements, which should correspond to the edges that are most relevant to the task. We achieve this regularization by including an ℓ_1-norm of the affinity matrix in the loss function.

3.4. Network Architecture

As illustrated in Fig. 3, the input of our Modulated GCN is 2D keypoints which can be obtained via an off-the-shelf 2D detector. Motivated by Martinez *et al*. [31], we use two modulated graph convolutional layers as a building block and the skip connection is applied. All the graph convolutional layers are followed by batch normalization [14] and a ReLU [34] activation function except for the last one. The 3D pose is generated by the last layer of the network. We use the weighted summation of an ℓ_2-norm loss and an ℓ_1-norm loss to compare the prediction and ground truth, *i.e*., both losses are imposed on the prediction error, and their weights are respectively 0.9 and 0.1.

4. Experiments

We first introduce experimental settings, evaluation metrics and implementation details in Sec. 4.1. The results of ablation study on each component of the proposed approach are reported in Sec. 4.2. We compare our Modulated GCN with state-of-the-art methods in Sec. 4.3. Finally, some qualitative results are shown in Sec. 4.4.

4.1. Setting

Dataset. We evaluate our approach on two standard benchmarks: Human3.6M [15] and MPI-INF-3DHP [33]. Human3.6M is the most widely used dataset in the 3D HPE literature. It contains 3.6 million images which are filmed by 4 synchronized cameras in different views. There are 15 daily activities such as walking, phoning, sitting and engaging in a discussion, performed by 11 human subjects in an indoor environment. The annotations of accurate 3D body joint coordinates are captured by a motion capture system, while the 2D poses are obtained by projection with known intrinsic and extrinsic camera parameters. Following previous work [31], we use standard normalization to preprocess the 2D and 3D poses before feeding them into our model. The hip joint is adopted as the root joint of 3D pose for zero-centering. MPI-INF-3DHP is a recent 3D human pose dataset constructed by a motion capture system with both indoor scenes and complex outdoor scenes. In contrast to Human3.6M, it covers more action classes ranging from walking and sitting to challenging exercise poses and dynamic actions. To demonstrate the generalization ability of our model quantitatively, we evaluate our model trained on Human3.6M on the testing set of MPI-INF-3DHP. The test split is made up of approximately 3k images from six subjects performing seven actions.

Evaluation protocols. Two standard protocols are exploited to evaluate our model on Human3.6M. We uses five subjects (S1, S5, S6, S7 and S8) for training and two subjects (S9 and S11) for testing under both Protocol #1 and Protocol #2. All the camera views are trained with a single model for all actions. Following previous work [31, 3, 6, 54, 9], two metrics are utilized to evaluate our approach on Human3.6M. The metric used in Protocol #1 is the mean per-joint position error (MPJPE) which measures the average euclidean distance in millimeter between the ground truth and the prediction after aligning the root joint (the hip joint). Another metric is the mean per-joint position error after Procrustes alignment (P-MPJPE), which is used in Protocol #2. This metric is invariant to both rotation and scaling. For MPI-INF-3DHP, a 3D extension of the Percentage of Correct Keypoints (3DPCK) and Area Under

Method	Channels	Params	MPJPE	P-MPJPE
$\mathbf{A}_{skeleton}$	128	0.27M	49.73	39.92
$\mathbf{A}_{mul}$	128	0.27M	43.05	33.43
$\mathbf{A}_{no-skeleton}$	128	0.27M	42.21	33.71
$\mathbf{A}_{mix}$	128	0.27M	41.10	32.02
$\mathbf{A}_{add}$	128	0.27M	**40.53**	**31.39**

Table 1. Ablation study on variants of affinity modulation. The units of MPJPE and P-MPJPE are millimeters (mm).

Method	Channels	Params	MPJPE	P-MPJPE
$\mathbf{A}_{add}$	128	0.27M	40.53	31.39
$\mathbf{A}_{add}$ + symmetry	128	0.27M	**39.42**	**31.08**
$\mathbf{A}_{add}$ + sparsity	128	0.27M	40.23	32.06
$\mathbf{A}_{add}$ + low-rank	128	0.27M	39.43	31.46
$\mathbf{A}_{add}$ + symmetry + ℓ_1-loss	128	0.27M	**38.52**	**31.06**

Table 2. Ablation study on imposing different regularizations on affinity modulation. The units of MPJPE and P-MPJPE are millimeters (mm). The ℓ_1-loss in the last row means to include an ℓ_1-norm loss of the prediction error as discussed in Sec. 3.4.

Method	Channels	Params	MPJPE	P-MPJPE	Infer. Time
Weight sharing	128	0.27M	40.53	31.39	0.008s
Weight modulation	124	0.27M	38.83	30.35	0.008s
Weight unsharing	128	4.22M	38.08	29.96	0.032s
Weight sharing	256	1.06M	39.39	31.24	0.008s
Weight modulation	256	1.10M	**37.43**	**29.73**	0.008s
Weight unsharing	256	16.83M	39.09	30.32	0.035s

Table 3. Ablation study on the proposed weight modulation. The units of MPJPE and P-MPJPE are millimeters (mm). All the three variants use the affinity modulation $\mathbf{A}_{add}$ defined in Eq.(7) without imposing regularizations. Infer. Time indicates (per-batch) inference time.

Method	Channels	Params	MPJPE	P-MPJPE
SemGCN	128	0.27M	42.14	33.53
SemGCN w/ Non-local [49]	128	0.43M	40.78	31.46
Modulated GCN	128	0.29M	**38.25**	**30.06**

Table 4. Comparison between the SemGCN [57] and the proposed Modulated GCN. The units of MPJPE and P-MPJPE are millimeters (mm).

the Curve (AUC) are adopted as the evaluation metrics.

Implementation details. Following previous work [38], we obtain 2D pose detections using the cascaded pyramid network (CPN) [27]. Our model is implemented in Pytorch and optimized via Adam [17]. All experiments are conducted on a single NVIDIA RTX 2080 Ti GPU. We initialize the weights in Modulated GCN using the technique described in [12]. 3D pose regression from 2D detections is more challenging than that from 2D ground truth as the former needs to deal with some extra uncertainty in the 2D space. Therefore, it is favorable to set different configurations for them to avoid overfitting and achieve better convergence. For 2D ground truth, we set the initial learning rate 0.001, the decay factor 0.96 per 4 epoch, the batch size 64. For 2D pose detections, we set the initial learning rate 0.005, the decay factor 0.65 per 4 epoch, the batch size 256. We also set the channels to 384 to handle the detection errors. Following [3], we incorporate a non-local layer [49] and a pose refinement module to improve the performance. In the ablation study, the pose refinement module and the non-local layer are excluded. Also, we use the 2D ground truth as input to bypass the influence from 2D pose detectors. When comparing with the states of the art, we use the Modulated GCN which combines weight modulation in Eq. (5) and the affinity modulation $\mathbf{A}_{add}$ in Eq. (7) respectively, and the symmetry regularization is applied to the affinity matrix. Previous works [28, 57, 53] find that decoupling the transformations of self-connections and other edges can significantly improve the 3D HPE performance, which we also observe. Therefore, we report results obtained by the decoupling versions of all GCN variants (detailed formulations are included in the supplementary material).

4.2. Ablation Study

We conduct comprehensive ablation study on Human3.6M. The proposed Modulated GCN contains two main components: weight modulation and affinity modulation. The objective is to validate the effectiveness of each component under controlled settings. Note that the 2D ground truth is taken as input to eliminate the extra uncertainty from the 2D pose detector.

Variants of Affinity Modulation. We investigate four different affinity modulation methods described in Sec. 3.3. We use the vanilla GCN supervised by an ℓ_2-norm loss of the prediction error as a base network and compare the performance obtained by different variants of affinity modulation. Note weight modulation is not included in this ablation study. The results are reported in Tab. 1. The observation that $\mathbf{A}_{add}$ and $\mathbf{A}_{mix}$ outperform $\mathbf{A}_{mul}$ indicates that learning edges beyond the skeleton graph is helpful. In addition, $\mathbf{A}_{add}$ outperforms $\mathbf{A}_{no-skeleton}$. This indicates it is important to enforce the skeleton prior in affinity modulation. Finally, mixing $\mathbf{A}_{mul}$ and $\mathbf{A}_{add}$, *i.e.*, $\mathbf{A}_{mix}$, performs worse than $\mathbf{A}_{add}$. The predefined graph $\mathbf{A}_{skeleton}$ performs much worse than learnable graphs due to its inflexibility.

Regularizations of Affinity Modulation. We evaluate the variants of regularizations discussed in Sec. 3.3. The vanilla GCN with affinity modulation $\mathbf{A}_{add}$ is taken as a baseline. The results are shown in Tab. 2. For the sparsity regularization, we have tried different regularization

Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Hossain & Little [39] (†)	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
Lee *et al*. [23] (†)	40.2	49.2	47.8	52.6	50.1	75.0	50.2	43.0	55.8	73.9	54.1	55.6	58.2	43.3	43.3	52.8
Cai *et al*. [3] (†)	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
Pavllo *et al*. [38] (†)	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
Xu *et al*. [52] (†)	37.4	43.5	42.7	42.7	46.6	59.7	41.3	45.1	52.7	60.2	45.8	43.1	47.7	33.7	37.1	45.6
Liu *et al*. [30] (†)	41.8	44.8	41.1	44.9	47.4	54.1	43.4	42.2	56.2	63.6	45.3	43.5	45.3	31.3	32.3	45.1
Wang *et al*. [48] (†)	40.2	42.5	42.6	41.1	46.7	56.7	41.4	42.3	56.2	60.4	46.3	42.2	46.2	31.7	31.0	**44.5**
Martinez *et al*. [31]	51.8	56.2	58.1	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
Sun *et al*. [43]	52.8	54.8	54.2	54.3	61.8	67.2	53.1	53.6	71.7	86.7	61.5	53.4	61.6	47.1	53.4	59.1
Yang *et al*. [54]	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
Fang *et al*. [9]	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
Pavlakos *et al*. [36]	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
Zhao *et al*. [57]	47.3	60.7	51.4	60.5	61.1	49.9	47.3	68.1	86.2	55.0	67.8	61.0	42.1	60.6	45.3	57.6
Sharma *et al*. [40]	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
Ci *et al*. [6]	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
Cai *et al*. [3] (single-frame)	46.5	48.8	47.6	50.9	52.9	61.3	48.3	45.8	59.2	64.4	51.2	48.4	53.5	39.2	41.2	50.6
Pavllo *et al*. [38] (single-frame)	47.1	50.6	49.0	51.8	53.6	61.4	49.4	47.4	59.3	67.4	52.4	49.5	55.3	39.5	42.7	51.8
Liu *et al*. [28] (weight unsharing)	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
Ours	45.4	49.2	45.7	49.4	50.4	58.2	47.9	46.0	57.5	63.0	49.7	46.6	52.2	38.9	40.8	**49.4**

Table 5. Quantitative comparisons on Human3.6M under Protocol #1. Errors are in millimeters. (†): uses temporal information.

Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Hossain & Little [39] (†)	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
Lee *et al*. [23] (†)	34.9	35.2	43.2	42.6	46.2	55.0	37.6	38.8	50.9	67.3	48.9	35.2	50.7	31.0	34.6	43.4
Cai *et al*. [3] (†)	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
Pavllo *et al*. [38] (†)	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
Xu *et al*. [52] (†)	31.0	34.8	34.7	34.4	36.2	43.9	31.6	33.5	42.3	49.0	37.1	33.0	39.1	26.9	31.9	36.2
Wang *et al*. [48] (†)	31.8	34.3	35.4	33.5	35.4	41.7	31.1	31.6	44.4	49.0	36.4	32.2	35.0	24.9	23.0	**34.5**
Martinez *et al*. [31]	39.5	43.2	46.4	47.0	51.0	56.0	41.4	40.6	56.5	69.4	49.2	45.0	49.5	38.0	43.1	47.7
Sun *et al*. [43]	42.1	44.3	45.0	45.4	51.5	53.0	43.2	41.3	59.3	73.3	51.0	44.0	48.0	38.3	44.8	48.3
Fang *et al*. [9]	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*. [36]	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
Li *et al*. [25]	35.5	39.8	41.3	42.3	46.0	48.9	36.9	37.3	51.0	60.6	44.9	40.2	44.1	33.1	36.9	42.6
Ci *et al*. [6]	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
Cai *et al*. [3] (single-frame)	36.8	38.7	38.2	41.7	40.7	46.8	37.9	35.6	47.6	51.7	41.3	36.8	42.7	31.0	34.7	40.2
Pavllo *et al*. [38] (single-frame)	36.0	38.7	38.0	41.7	40.1	45.9	37.1	35.4	46.8	53.4	41.4	36.9	43.1	30.3	34.8	40.0
Liu *et al*. [28] (weight unsharing)	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
Ours	35.7	38.6	36.3	40.5	39.2	44.5	37.0	35.4	46.4	51.2	40.5	35.6	41.7	30.7	33.9	**39.1**

Table 6. Quantitative comparisons on Human3.6M under Protocol #2. Errors are in millimeters. (†): uses temporal information.

weights, *i.e*., 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, and reported the best result (achieved at 10^{-6}). For the low-rank regularization, we have tried different subspace dimensions, *i.e*., 7, 9, 11, and reported the best result (achieved at 9). We can see that the baseline is enhanced by imposing the symmetry regularization on the affinity matrix. Specifically, it reduces MPJPE by 1.11mm. However, the sparsity and low-rank regularizations are not as effective as the symmetry regularization. We have also tried the combination of sparsity and symmetry regularizations and observed degraded performance. Note the low-rank regularization automatically imposes symmetry. Finally, we replace the widely-used ℓ_2-norm loss with a combination of ℓ_2-norm and ℓ_1-norm losses imposed on the prediction error (as discussed in Sec. 3.4) and see that this can further improve the performance.

Weight Modulation. The proposed weight modulation module aims to address the limitation of the vanilla GCN which assigns a shared feature transformation for all nodes and the inefficiency of weight unsharing GCNs in Eq. (3) caused by completely unsharing feature transformations for different nodes. We use the vanilla GCN with affinity modulation $\mathbf{A}_{add}$ as a baseline, and replace weight sharing with

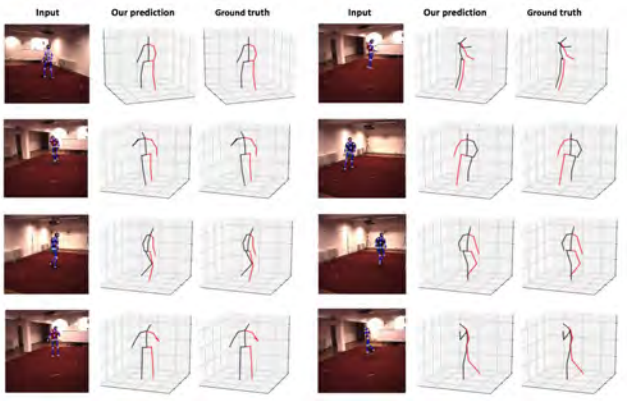

Figure 4. Qualitative results obtained by our Modulated GCN on the Human3.6M dataset.

Methods	GS	no GS	Outdoor	3DPCK	AUC
Mehta *et al.* [33]	70.8	62.3	58.5	64.7	31.7
Zhou *et al.* [59]	75.6	71.3	80.3	75.3	38.0
Zhou *et al.* [60]	71.1	64.7	72.7	69.2	32.5
Yang *et al.* [54]	-	-	-	69.0	32.0
Pavlakos *et al.* [36]	76.5	63.1	77.5	71.9	35.3
Wang *et al.* [47]	-	-	-	71.9	35.8
Martinez *et al.* [31]	49.8	42.5	31.2	42.5	17.0
Li *et al.* [25]	70.1	68.2	66.6	67.9	-
Liu *et al.* [28] (weight unsharing)	77.6	80.5	80.1	79.3	47.6
Ours	**86.4**	**86.0**	**85.7**	**86.1**	**53.7**

Table 7. Quantitative comparisons on MPI-INF-3DHP. GS denotes green screen. A higher value of 3DPCK or AUC indicates better performance.

weight modulation and weight unsharing. The results are shown in Tab. 3. We can see that weight modulation has comparable performance with [28] by using only 0.27M parameters, a 93.6% reduction of model size. Doubling channels, it outperforms [28] by using 1.10M parameters, a 73.9% reduction of model size.

Tab. 4 further compares our Modulated GCN with the Semantic GCN (SemGCN) [57], a state-of-the-art variant of GCN designed for 2D-to-3D pose lifting. To eliminate the influence from the 2D pose detector, we report results on 2D ground truth. We can see that our Modulated GCN can outperform the SemGCNs with and without non-local modules. Note our Modulated GCN does not use non-local modules. This demonstrates the great advantage of our Modulated GCN. More comparison between the proposed approach and state-of-the-art GCNs can be found in Fig. 1. We can see the Modulated GCN achieves the best trade-off between performance and the model size.

4.3. Comparison with State of the Art

Human3.6M. We compare the Modulated GCN with some state-of-the-art methods on Human3.6M under both Protocol #1 and Protocol #2. Following previous works [38, 3, 28], we use 2D poses detected by a pre-trained cascaded pyramid network (CPN) [27] as input. The results are reported in Tab. 5 and Tab. 6. Note that some approaches [38, 3, 52, 30, 23] exploit temporal smoothness by taking monocular video clips as input. However, our Modulated GCN is still very competitive and outperforms all the other methods except for those using temporal information.

MPI-INF-3DHP. We evaluate our Modulated GCN on the testing set of MPI-INF-3DHP to test its generalization ability across different datasets. Following [25], we use the 2D joints provided by the dataset as input. The results are shown in Tab. 7. Our approach shows superior performance over all the other methods under both the indoor and outdoor scenes. Note that our model is only trained with indoor scenes on Human3.6M, but it achieves satisfactory results on outdoor scenes. This indicates that our model generalizes well to unseen actions and datasets.

4.4. Qualitative Results

Fig. 4 shows some visualization results obtained by our Modulated GCN on Human3.6M. It can accurately predict 3D poses of different persons who are performing various actions. The difference between our prediction and the ground truth is usually negligible. Some 2D pose estimations are not perfect especially when occlusion happens. But reasonable predictions can still be generated by our approach.

5. Conclusions

From extensive ablation study and benchmark experiments, we draw the following conclusions. (1) Weight modulation can resolve the limitation caused by a shared feature transformation for different nodes while retaining a small size. (2) Affinity modulation is necessary for good performance. Both the skeleton prior and the ability to learn edges beyond the skeleton are important for affinity modulation. (3) Affinity modulation with an unconstrained modulation matrix does not cause severe generalization problems. Only the symmetry regularization can improve its performance. (4) The Modulated GCN, integrating weight modulation and affinity modulation, achieves the best trade-off between performance and the model size among all GCN approaches. (5) The Modulated GCN outperforms some recent states of the art on two benchmark datasets.

Acknowledgements. This work was supported in part by Wei Tang's startup funds from the University of Illinois at Chicago and the National Science Foundation (NSF) award CNS-1828265.

References

[1] James Atwood and Don Towsley. Diffusion-convolutional neural networks. In *Advances in neural information processing systems*, pages 1993–2001, 2016.

[2] Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[3] Yujun Cai, Liuhao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2272–2281, 2019.

[4] Zerui Chen, Yan Huang, Hongyuan Yu, Bin Xue, Ke Han, Yiru Guo, and Liang Wang. Towards part-aware monocular 3d human pose estimation: An architecture search approach. 2020.

[5] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. *arXiv preprint arXiv:2008.09047*, 2020.

[6] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2262–2271, 2019.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[8] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[9] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[10] Junyu Gao, Tianzhu Zhang, and Changsheng Xu. Graph convolutional tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4649–4659, 2019.

[11] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[13] Ankur Gupta, Julieta Martinez, James J Little, and Robert J Woodham. 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2601–2608, 2014.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[16] Hao Jiang. 3d human pose reconstruction using millions of exemplars. In *2010 20th International Conference on Pattern Recognition*, pages 1674–1677. IEEE, 2010.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[21] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[22] Hsi-Jian Lee and Zen Chen. Determination of 3d human body postures from a single view. *Computer Vision, Graphics, and Image Processing*, 30(2):148–168, 1985.

[23] Kyoungoh Lee, Inwoong Lee, and Sanghoon Lee. Propagating lstm: 3d pose estimation based on joint interdependency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–135, 2018.

[24] Ron Levie, Federico Monti, Xavier Bresson, and Michael M Bronstein. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Transactions on Signal Processing*, 67(1):97–109, 2018.

[25] Chen Li and Gim Hee Lee. Generating multiple hypotheses for 3d human pose estimation with mixture density network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9887–9895, 2019.

[26] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10313–10322, 2019.

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[28] Kenkun Liu, Rongqi Ding, Zhiming Zou, Le Wang, and Wei Tang. A comprehensive study of weight sharing in graph networks for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[29] Kenkun Liu, Zhiming Zou, and Wei Tang. Learning global pose features in graph convolutional networks for 3d human pose estimation. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[30] Ruixu Liu, Ju Shen, He Wang, Chen Chen, Sen-ching Cheung, and Vijayan Asari. Attention mechanism exploits temporal contexts: Real-time 3d human pose reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[31] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017.

[32] Andreas Maurer, Massimiliano Pontil, and Gabor Lugosi. Structured sparsity and generalization. *Journal of Machine Learning Research*, 13(3), 2012.

[33] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *2017 International Conference on 3D Vision (3DV)*, pages 506–516. IEEE, 2017.

[34] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[35] Sungheon Park, Jihye Hwang, and Nojun Kwak. 3d human pose estimation using convolutional neural networks with 2d pose information. In *European Conference on Computer Vision*, pages 156–169. Springer, 2016.

[36] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7307–7316, 2018.

[37] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017.

[38] Dario PavlloF, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[39] Mir Rayat Imtiaz Hossain and James J Little. Exploiting temporal information for 3d human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68–84, 2018.

[40] Saurabh Sharma, Pavan Teja Varigonda, Prashast Bindal, Abhishek Sharma, and Arjun Jain. Monocular 3d human pose estimation by generation and ordinal ranking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2325–2334, 2019.

[41] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019.

[42] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

[43] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2602–2611, 2017.

[44] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018.

[45] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*, 2016.

[46] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[47] Jue Wang, Shaoli Huang, Xinchao Wang, and Dacheng Tao. Not all parts are created equal: 3d pose estimation by modeling bi-directional dependencies of body parts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7771–7780, 2019.

[48] Jingbo Wang, Sijie Yan, Yuanjun Xiong, and Dahua Lin. Motion guided 3d pose estimation from videos. In *European Conference on Computer Vision*, pages 764–780. Springer, 2020.

[49] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

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

[51] Hang Xu, ChenHan Jiang, Xiaodan Liang, and Zhenguo Li. Spatial-aware graph relation network for large-scale object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9298–9307, 2019.

[52] Jingwei Xu, Zhenbo Yu, Bingbing Ni, Jiancheng Yang, Xiaokang Yang, and Wenjun Zhang. Deep kinematics analysis for monocular 3d human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 899–908, 2020.

[53] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[54] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5255–5264, 2018.

[55] Ailing Zeng, Xiao Sun, Fuyang Huang, Minhao Liu, Qiang Xu, and Stephen Lin. Srnet: Improving generalization in 3d human pose estimation with a split-and-recombine approach. *arXiv preprint arXiv:2007.09389*, 2020.

[56] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):11, 2019.

[57] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N Metaxas. Semantic graph convolutional networks for

3d human pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3425–3435, 2019.

[58] Rui Zhao, Kang Wang, Hui Su, and Qiang Ji. Bayesian graph convolution lstm for skeleton based action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6882–6892, 2019.

[59] Kun Zhou, Xiaoguang Han, Nianjuan Jiang, Kui Jia, and Jiangbo Lu. Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose stimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2344–2353, 2019.

[60] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 398–407, 2017.

[61] Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. Deep kinematic pose regression. In *European Conference on Computer Vision*, pages 186–201. Springer, 2016.

[62] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4966–4975, 2016.

[63] Zhiming Zou, Kenkun Liu, and Wei Tang. High-order graph convolutional networks for 3d human pose estimation.