High-order Graph Convolutional Networks for 3D Human Pose Estimation

Zhiming Zou[1]
zzou6@uic.edu
Kenkun Liu[1]
kliu44@uic.edu
Le Wang[2]
lewang@xjtu.edu.cn
Wei Tang*[1]
tangw@uic.edu

[1] University of Illinois at Chicago
Chicago, IL, USA
[2] Xi'an Jiaotong University
Xi'an, Shaanxi, P.R. China

Abstract

Graph convolutional networks (GCNs) have been applied to 3D human pose estimation (HPE) from 2D body joint detections and have demonstrated promising performance. However, since the vanilla graph convolution is performed on the one-hop neighbors of each node, it is unable to capture the long-range dependencies between body joints. They can help reduce the uncertainty caused by occlusion or depth ambiguity. To resolve this issue, we propose a high-order GCN for 3D HPE. Its core building block, termed a high-order graph convolution, aggregates features of nodes at various distances. As a result, the network can model a wide range of interactions among body joints. Furthermore, we investigate different methods to fuse those multi-order features and compare how they affect the performance. Experimental results demonstrate the effectiveness of the proposed approach.

1 Introduction

3D human pose estimation (HPE) aims to predict the 3D locations of body joints in the camera coordinate system from a monocular image. It is a fast-growing research area and has attracted extensive attention in the computer vision community due to its numerous real-world applications such as human-computer interaction, action recognition, video synthesis, and motion capture. However, 3D HPE remains a challenging problem especially as multiple valid 3D poses can be projected to the same 2D pose in the image space.

The state-of-the-art 3D HPE systems are built on deep neural networks [23] due to their strong capability to learn powerful feature representations. Some approaches [34, 41, 42, 48, 54] directly regress the 3D pose via a convolutional neural network [22, 24] from an image and demonstrate superior performance over earlier methods relying on handcrafted features [2, 17, 38]. Other works formulate the problem as 2D keypoint detection [9, 16, 43, 44] followed by 2D-to-3D pose lifting [6, 10, 30, 36, 50]. For example, Martinez *et al*. [30] use

* Corresponding author.

a simple fully connected network with only 2D keypoints detection as input and achieve the state-of-the-art 3D HPE performance.

Some recent approaches [6, 10, 50)] exploit graph convolutional networks (GCNs) [9, 14, 21] to model the relationships between neighboring body joints and demonstrate their superiority over the fully connected networks. A GCN consists of multiple graph convolution layers and repeatedly transforms and aggregates features of neighboring nodes to get increasingly more powerful representations. However, one potential limitation of existing GCNs designed for 3D HPE is that they perform graph convolutions only on the one-hop neighbors of each node. As a result, they are unable to capture the long-range dependencies between body joints, which can be critical to reduce the uncertainty caused by occlusion or depth ambiguity.

To address this problem, this paper introduces a high-order GCN for 3D HPE. Its core building block, termed a high-order graph convolution, aggregates features of nodes at various distances, which enables the model to learn a wide range of interactions among body joints. It is easy to find the k-hop neighbors of each node in a graph by computing the kth power of the adjacency matrix. However, one critical problem in designing the high-order GCN is how to fuse the features of these multi-hop neighbors. The most simple strategy is to connect the distant nodes directly on the graph, which is equivalent to summing up the adjacency matrix up to its kth power. Unfortunately, naively modifying the graph structure degrades the performance possibly because the model cannot distinguish neighbors at different hops. Thus, we investigate two alternative fusion strategies. Specifically, we transform the features of nodes at different distances separately and then aggregate them via summation or concatenation. Extensive ablation study shows that (1) the fusion method has a significant impact on the performance of high-order GCNs and the concatenation-based approach leads to the best performance and (2) the high-order GCN outperforms the vanilla GCN, which demonstrates the importance of modeling long-range relationships among body joints as well as the effectiveness of the proposed approach.

In sum, the contribution of this paper is threefold.

- We introduce high-order GCNs for 3D HPE. They can learn long-range dependencies among body joints, which is critical to resolve the uncertainty caused by occlusion or depth ambiguity.
- We investigate three strategies to fuse the features of multi-hop neighbors and show that it is critical to choose the optimal strategy to achieve the best performance.
- We conduct extensive ablation study to compare the high-order GCNs and the vanilla GCN as well as different feature fusion methods. Experimental results demonstrate that the proposed approach can outperform state-of-the-art methods.

2 Related Work

2.1 3D Human Pose Estimation

The problem of predicting 3D poses from images can be dated back to Lee and Chen [25]. A standard method is to predict the 2D poses first, and use them to infer the 3D poses by K-Nearest Neighbor [7, 19, 49]. Recently, state-of-the-art 3D HPE approaches take advantage of deep neural networks, and they can be roughly divided into two categories.

The first category of approaches mainly exploit convolutional neural networks (CNNs) to obtain the 3D pose directly from the input image [31, 33, 34, 42, 53, 54]. Some of them tend to learn robust and powerful representations. For example, Zhou *et al*. [54] integrate a 3D depth regression sub-network into a state-of-the-art 2D detector. Pavlakos *et al*. [34] propose a fine discretization of the 3D space around the subject and train a CNN to predict the per voxel likelihood for each body joint. Sun *et al*. [42] design a simple integral operation to relate and unify the heat map representation and joint regression. Some other approaches incorporate 3D geometry prior to deep learning. Zhou *et al*. [53] train a deep neural network with a kinematic object model embedding into it for general articulated object pose estimation. Zhou *et al*. [52] utilize a sparsity-driven 3D geometric prior and temporal smoothness to regress 3D poses from uncertain 2D keypoints maps via the EM algorithm.

The second category of approaches formulate the task of 3D HPE into two subtasks [5, 30, 32, 50]. An off-the-shelf 2D pose detector first obtains the coordinates of 2D body joints from the input image. Then, they are passed to a neural network for 3D pose regression. Our approach belongs to this family. The work most related to ours are [6, 10, 29, 50] as they also rely on graph convolutional networks (GCNs). Cai *et al*. [6] use graph pooling and upsampling techniques to build a local-to-global network and expand the graph convolution as a summation of multiple kernels corresponding to different semantic meanings. Zhao *et al*. [50] propose a semantic GCN by multiplying a learnable mask to the affinity matrix and applying different weights to each output channel. Ci *et al*. [10] introduce a locally connected network to enhance the representation capability of GCN. Liu *et al*. [29] have a comprehensive investigation of weight sharing in a GCN. Our proposed high-order GCN differs from previous methods in that it aggregates features of body joints at various distances via mixing powers of the adjacency matrix, which boosts the representation capability of GCNs by capturing long-range dependencies among body joints. Also, we explore the optimal way to fuse the multi-order feature representation.

2.2 Graph Convolutional Networks

GCNs [9, 11, 14, 21] generalize convolutional neural networks by performing convolutions on graph data. There are roughly two types of GCNs depending on whether they are constructed from a spectral [11, 26, 40] or spatial perspective [9, 14, 21, 45]. The proposed high-order GCN performs convolutions directly on the graph nodes and their neighbors, which is more related to spatial GCNs. GCNs, as an effective alternative of CNNs, have been applied to other computer vision tasks, e.g., action recognition [51], visual question answering [27], object detection [47], tracking [13], multi-label image recognition [8].

The work most related to ours is Sami *et al*. [1], which designs a higher-order GCN to mix feature representations of neighbors at various distances. Our work is different from them in three aspects. First, we focus on 3D human pose estimation which is a regression task, while their task is node classification. Second, they only explore a shallow network with two layers and limit the network to two-hop message passing. By contrast, we use a much deeper network and compare GCNs involving different hops of neighbors. Last but not least, we explore different ways to fuse the multi-order feature representations, which they ignore. Bai *et al*. [3] exploit a high-order GCN for skeleton-based action recognition, but their high-order adjacency matrix is constructed via summing up the mixed powers of the original adjacency matrix. We will discuss the limitation of this simple method and show it leads to inferior 3D HPE performance.

3 Our Approach

We first revisit the graph convolutional network [21] (Sec. 3.1). The high-order GCN is proposed to learn long-range dependencies among body joints (Sec. 3.2). Finally, we show our network architecture in detail (Sec. 3.3).

3.1 Revisit GCN

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ denote a graph where $\mathcal{V}$ is a set of N nodes and $\mathcal{E}$ is the collection of all edges. The edges can be encoded via an adjacency matrix $\mathbf{A} \in \{0,1\}^{N\times N}$. Each node i is associated with a D-dimensional feature vector $\mathbf{x}_i \in R^D$. The collection of all feature vectors can be written as a matrix $\mathbf{X} \in R^{D\times N}$ where the ith column of $\mathbf{X}$ is $\mathbf{x}_i$. A graph convolution layer updates the features of each node via the equation below:

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

where $\tilde{\mathbf{A}}$ is the symmetrically normalized version of $\mathbf{A}$ with self-connections [21], $\mathbf{W} \in R^{D'\times D}$ is a learnable weight matrix transforming the feature dimension from D to D', $\sigma(\cdot)$ is an activation function, $\mathbf{X}' \in R^{D'\times N}$ is the updated feature matrix. A GCN consists of multiple graph convolution layers that repeatedly transform and aggregate features of neighboring nodes to get increasingly more powerful representations, which are used by the last layer to predict the output.

We empirically find that decoupling the transformations for the self-nodes and the 1-hop neighbors can significantly improve the performance of 3D HPE:

$$\mathbf{X}' = \sigma(\mathbf{W}^{(0)}\mathbf{X} + \mathbf{W}^{(1)}\mathbf{X}\hat{\mathbf{A}}) \tag{2}$$

where $\mathbf{W}^{(0)} \in R^{D'\times D}$ and $\mathbf{W}^{(1)} \in R^{D'\times D}$ are the weight matrices corresponding to the self and neighbor transformations respectively, $\hat{\mathbf{A}}$ is the symmetrically normalized version of $\mathbf{A}$ without self-connections. We will take Eq. (2) as a strong baseline in the experiments.

3.2 High-order GCN

As shown in Fig. 1(a), the graph convolutions defined in Eqs. (1) and (2) only focus on the 1-hop neighbors, which limits their ability to capture the long-range dependencies among nodes. To address this problem, we propose high-order graph convolutions to take into account multi-hop neighbors when updating the node features, which is illustrated in Fig. 1(b).

If $\mathbf{A}$ is the adjacency matrix of a graph, the element (i, j) of the matrix $\mathbf{A}^k$ (i.e., the matrix product of k copies of $\mathbf{A}$) is nonzero if and only if the nodes i and j are k-hop neighbors of each other. Note $\mathbf{A}^0$ is an identity matrix, which indicates the self-connections widely used in the conventional GCNs can be considered as the 0-hop neighbors. Thus, a naive way to capture the multi-hop neighbors in a graph convolution is to sum up the mixed powers of the adjacency matrix and use it in the original graph convolution:

$$\mathbf{X}' = \sigma(\mathbf{W}\mathbf{X}\sum_{k=0}^{K}\hat{\mathbf{A}}^k) \tag{3}$$

where K is the maximum order of the neighbors to be involved. $\hat{\mathbf{A}}^k$ first raises $\mathbf{A}$ to its kth power and then applies the symmetrical normalization. This formulation means to merge the

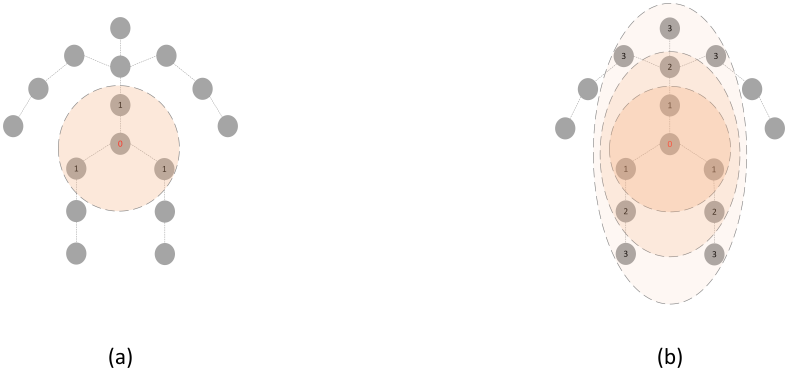

Figure 1: Comparison between the vanilla graph convolution and the high-order graph convolution performed on a skeleton graph. The number $k \in \{0,1,2,3\}$ in a node indicates the corresponding body joint is a k-hop neighbor of the pelvis. The range of dependencies modeled by each graph convolution is represented by the orange ellipse. (a) A vanilla graph convolution only focuses on the 1-hop neighbors. (b) A high-order graph convolution takes into account neighbors at different distances.

multi-order relationships among edges into a single adjacency matrix, which is equivalent to modifying the graph structure by connecting the distant nodes directly on the graph. A potential problem of this strategy is that it turns the indirect relationships between distant body joints into direct ones.

As a result, it inherits the drawback of the graph convolution defined in Eq. (1) that a shared weight matrix is used to transform the features of all neighbors at different distances. Actually, Eq. (1) can be considered as a special case of Eq. (3) with $K = 1$. Our experimental results demonstrate that this kind of oversimplified multi-hop modeling leads to inferior performance.

Thus, we propose the following alternative form of a high-order graph convolution:

$$\mathbf{X}^{'} = \sigma(\mathcal{F}(\{\mathbf{W}^{(k)}\ \mathbf{X}\hat{\mathbf{A}}^{k} : k = 0, \cdots, K\})) \tag{4}$$

$\mathcal{F}$ is a fusion function, $\mathbf{W}^{(k)}$ is the weight matrix corresponding to the k-hop neighbors. Eq. (4) means to transform and aggregate node features at different hops via *unshared* weight matrices and then fuse them, which can address the limitation of Eq. (3). We consider two fusion functions widely used in deep learning: summation and concatenation. After instantiating the fusion function $\mathcal{F}$ as a summation function, the high-order graph convolution can be rewritten as

$$\mathbf{X}^{'} = \sigma(\sum_{k=0}^{K} \mathbf{W}^{(k)}\ \mathbf{X}\hat{\mathbf{A}}^{k}) \tag{5}$$

It assigns a different weight matrix $\mathbf{W}^{(k)}$ to neighbors at different hops and fuses the features via summation. Note the graph convolution defined in Eq. (2) can be considered as a special case of Eq. (5) when $K = 1$.

Alternatively, we can instantiate the fusion function $\mathcal{F}$ as a concatenation function:

$$\mathbf{X}^{'} = \sigma(\mathrm{Cat}(\mathbf{W}^{(0)}\ \mathbf{X}\hat{\mathbf{A}}^{0}, \cdots, \mathbf{W}^{(K)}\ \mathbf{X}\hat{\mathbf{A}}^{K})) \tag{6}$$

where the concatenation occurs on the channel dimension.

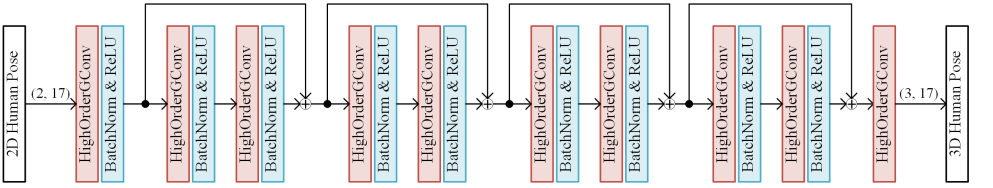

Figure 2: An example high-order GCN for 3D human pose estimation. 17 is the number of body joints.

3.3 Network Architecture

As illustrated in Fig. 2, the input of a high-order GCN is the 2D coordinates of body joints in the image space. Inspired by Martinez *et al*. [30], we use the residual block consisting of two *HighOrderGConv* layers as a building block and repeat it several times. All *HighOrderGConv* layers are followed by batch normalization and a ReLU activation except for the last one. The last *HighOrderGConv* outputs the 3D body joint coordinates in the camera coordinate system. An L2-norm loss is used to compare the 3D human pose prediction and the ground truth during training.

4 Experiments

4.1 Setting

Dataset. We evaluate our approach on the Human3.6M dataset [18]. It is the most widely used benchmark in the 3D HPE literature. Human3.6M consists of 3.6 million images which are taken from 4 synchronized cameras with different views. There are 15 daily activities (walking, eating, sitting, etc.) captured by 11 human subjects (5 females and 6 males) in an indoor environment. The 3D human pose is represented as the 3D coordinates of 17 body joints. The annotation includes precise 2D and 3D body joint coordinates as well as camera parameters. The ground truth are obtained by motion capture devices. Following previous work [30], we use standard normalization to preprocess the 2D and 3D poses before feeding them to our model. The hip joint is adopted as the root joint of 3D poses for zero-centering.

Evaluation protocols. The Human3.6M benchmark defines two protocols for evaluation. Protocol #1 uses five subjects (S1, S5, S6, S7 and S8) for training and two subjects (S9 and S11) for testing. Another protocol adopts six subjects S1, S5, S6, S7, S8 and S9 as the training set, and S11 is used as the testing set. We refer this as Protocol #2. Evaluation is performed on every 64th frame of the testing set. Following previous work, two metrics are utilized to evaluate our approach on Human3.6M. The metric applied 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 scale.

Implementation details. Following previous work [36], we use the cascaded pyramid network (CPN) [9] to extract the 2D poses from input images, and the pose bounding boxes are obtained by the Mask-RCNN [16] with a ResNet-101-FPN [28] backbone. Both the Mask-RNN and CPN (pre-trained on the COCO dataset) are fine-tuned on Human3.6M since

Method	Channels	# Params	MPJPE	P-MPJPE
2-hop A-summation	385	1.20M	43.69	35.64
2-hop feature-summation	223	1.20M	42.64	33.46
2-hop feature-concatenation	128	1.20M	**39.68**	**31.69**
3-hop A-summation	385	1.20M	45.18	34.95
3-hop feature-summation	193	1.20M	40.74	31.86
3-hop feature-concatenation	96	1.20M	**39.52**	**31.07**

Table 1: Ablation study on variants of high-order graph convolutions. The units of MPJPE and P-MPJPE are millimeters (mm).

Method	Channels	# Params	MPJPE	P-MPJPE
1-hop feature-concatenation	192	1.20M	42.99	34.67
2-hop feature-concatenation	128	1.20M	39.68	31.69
3-hop feature-concatenation	96	1.20M	**39.52**	**31.07**

Table 2: Ablation study on the impact of orders. The units of MPJPE and P-MPJPE are millimeters (mm).

the keypoints in COCO are different from those in Human3.6M. In our ablation study, the 2D ground truth is used as input to eliminate the influence of the 2D pose detector.

We implement our model in PyTorch and optimize it via Adam [20]. All experiments are conducted on a single NVIDIA RTX 2080 Ti GPU. We initialize the weights in high-order GCNs with the initialization technique described in [15]. Max-norm is used to keep the weights in each layer within [0, 1]. 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. We find it is beneficial to set different configurations for them to avoid overfitting and achieve better convergence. For the 2D ground truth, we set the initial learning rate 0.001, the decay factor 0.96 per 100,000 steps, the batch size 64. For 2D pose detections, we set the initial learning rate 0.005, the decay factor 0.8 per 100,000 steps, the batch size 256. In the ablation study, we test the impact of the order and fusion methods on the performance. When comparing with state-of-the-art methods, we use a three-hop high-order GCN with feature concatenation as the fusion method, i.e., Eq. (6).

4.2 Ablation Study

We conduct extensive ablation experiments on the Human3.6M dataset. The 2D ground truth is taken as input. The objective is to test the impact of the order and fusion methods on the performance. Specially, we denote the three variants of high-order graph convolutions defined in Eq. (3), Eq. (5) and Eq. (6) as *A-summation*, *feature-summation* and *feature-concatenation*, respectively. We use the graph convolution defined in Eq. (2) as our baseline GCN.

Variants of high-order graph convolutions. We compare the three strategies to model the multi-hop neighbors in a high-order graph convolution. The results on the Human3.6M dataset are shown in Tab. 1. We adjust the number of channels, i.e., the number of rows of $\mathbf{W}^{(k)}$, to control the size of all the models.We show that simply summing up the mixed powers of the adjacency matrix leads to the worst performance. The method of *feature-*

Method	Channels	# Params	MPJPE	P-MPJPE
baseline GCN	136	0.30M	41.79	33.55
3-hop feature-concatenation	48	0.30M	**40.77**	**31.88**
baseline GCN	273	1.20M	40.99	31.75
3-hop feature-concatenation	96	1.20M	**39.52**	**31.07**

Table 3: Comparison between the baseline GCN and the proposed high-order GCN. The units of MPJPE and P-MPJPE are millimeters (mm)

Method	# Params	Training time	Inference time
baseline GCN	1.20M	0.040s	0.008s
1-hop feature-concatenation	1.20M	0.040s	0.009s
2-hop feature-concatenation	1.20M	0.050s	0.011s
3-hop feature-concatenation	1.20M	0.060s	0.013s

Table 4: Comparison of (per-batch) training and inference time between the baseline GCN and the proposed high-order GCN.

Method	Model size	Example ($K=3, C=128$)
baseline GCN	$O(2C^2)$	0.27M
K-hop A-summation	$O(C^2)$	0.14M
K-hop feature-summation	$O((K+1)C^2)$	0.53M
K-hop feature-concatenation	$O((K+1)^2C^2)$	2.12M

Table 5: Comparison of model size between the baseline GCN and the proposed high-order GCN. C denotes channels, i.e., the number of rows of the weight matrix $\mathbf{W}^{(k)}$.

concatenation defined in Eq. (6) outperforms the other two methods by a large margin in both two-hop and three-hop cases. Thus, we will use *feature-concatenation* as the high-order graph convolution in the remaining experiments.

Impact of orders. We change the range of neighbors involved in the high-order GCN and show the results in Tab. 2. We can see that the 2-hop model reduces the MPJPE and P-MPJPE of its 1-hop counterpart by 3.31mm and 2.98mm respectively. The 3-hop model has slightly better performance than the 2-hop model. These results indicate that the proposed high-order GCN can effectively capture the long-range dependencies among body joints and improve the 3D HPE.

Comparison with the baseline. In Tab. 3, we compare our model with the baseline GCN defined in Eq. (2). We can see that our model outperforms the baseline regardless of the model size.

Impact of orders on training/inference time and model size. In Tab. 4, we investigate the impact of high-order relations on training and inference time. The number of parameters is fixed as 1.20M. The inclusion of high-order relations will increase the training and inference time as the processing of nodes of different orders, e.g., $\mathbf{W}^{(k)}\ \mathbf{X}\hat{\mathbf{A}}^k$ in Eq. (4), is implemented sequentially. In Tab. 5, we further study the influence of high-order relations on the model size. The model size depends on the number of rows of the weight matrix $\mathbf{W}^{(k)}$, i.e., channels in Tabs. 1-3, and we denote it by C here. The number of its columns is determined by the input of each layer.

Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Martinez *et al.* [34]	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.* [39]	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.* [43]	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.* [6]	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.* [51]	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.* [38]	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
Ours	49.0	54.5	52.3	53.6	59.2	71.6	49.6	49.8	66.0	75.5	55.1	53.8	58.5	40.9	45.4	**55.6**

Table 6: Quantitative comparisons on Human3.6M under Protocol #1. Errors are in millimeters.

Method	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Zhou *et al.* [50]	47.9	48.8	52.7	55.0	56.8	49.0	45.5	60.8	81.1	53.7	65.5	51.6	50.4	54.8	55.9	55.3
Pavlakos *et al.* [36]	47.5	50.5	48.3	49.3	50.7	55.2	46.1	48.0	61.1	78.1	51.1	48.3	52.9	41.5	46.4	51.9
Martinez *et al.* [34]	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.* [39]	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.* [6]	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
Hossain & Little [36]	35.7	39.3	44.6	43.0	47.2	54.0	38.3	37.5	51.6	61.3	46.5	41.4	47.3	34.2	39.4	44.1
Ours	38.6	42.8	41.8	43.4	44.6	52.9	37.5	38.6	53.3	60.0	44.4	40.9	46.9	32.2	37.9	**43.7**

Table 7: Quantitative comparisons on Human3.6M under Protocol #2. Errors are in millimeters.

4.3 Comparison with the State of the Art

We quantitatively compare our approach with some state-of-the-art methods on Human3.6M. The results are shown in Tabs. 6 and 7.

Note that many leading approaches, complementary to ours, have exploited ideas or strategies from which our high-order GCN can also benefit. For example, Sharma *et al.* [39] train a conditional variational autoencoder to generate 3D pose samples and use ordinal annotations. Some other methods [6, 36] focus on video-based 3D pose estimation. Our method does not outperform them in their single-frame settings. [36] uses a fully connected network, whose model size is nearly 7 times as large as ours. [6] uses several strategies to boost their performance, including data augmentation, an additional symmetric loss, non-local layers and an additional pose refinement network, and their model size is more than 3 times as large as ours. These strategies are complementary to our method and can be used to improve the performance.

Tab. 8 further compares our high-order GCN with the Semantic GCN (SemGCN) [50], 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 high-order GCN (3-hop *feature-concatenation*) can outperform the SemGCN (without non-local) by 2.62mm under Protocol #1 and 2.46mm under Protocol #2. Note the non-local module [46] is designed to capture the non-local relationships among nodes but SemGCN with non-local modules still performs worse than our approach. This demonstrates the great advantage of our high-order GCN.

4.4 Qualitative Results

Fig. 3 shows some qualitative results obtained by our high-order GCN on the Human3.6M dataset. Our high-order GCN can infer 3D poses from input images in various situations.

Method	MPJPE	P-MPJPE
SemGCN	42.14	33.53
SemGCN w/ non-local [46]	40.78	31.46
Ours	**39.52**	**31.07**

Table 8: Comparison between our high-order GCN and the Semantic GCN (SemGCN) [50] on Human3.6M. All models take 2D ground truth as input. Errors are in millimeters

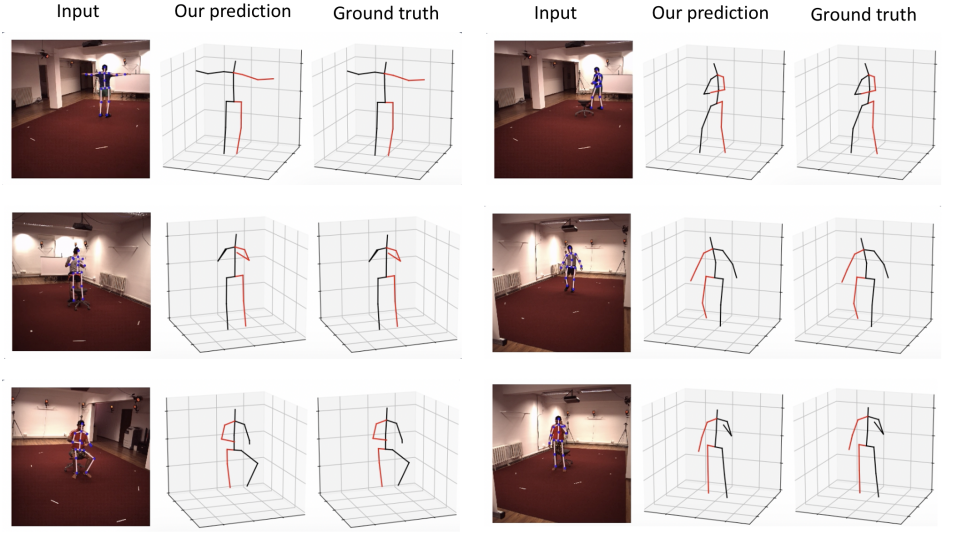

Figure 3: Qualitative results obtained by our high-order GCN on the Human3.6M dataset.

When the 2D detector fails due to self-occlusion, our model can provide plausible results.

5 Conclusion

In this paper, we introduce a conceptually simple but effective high-order graph convolutional network for 3D HPE. It learns a wide class of interactions among body joints and effectively captures the long-range dependencies between each body part and their distant neighbors. We also study different methods to fuse those multi-hop features. Experimental results demonstrate the effectiveness of the proposed approach.

Acknowledgments

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] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Hrayr Harutyunyan, Nazanin Alipourfard, Kristina Lerman, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolution architectures via sparsified neighborhood mixing. *arXiv preprint arXiv:1905.00067*, 2019.

[2] Ankur Agarwal and Bill Triggs. Recovering 3d human pose from monocular images. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):44–58, 2005.

[3] Zhimin Bai, Hongping Yan, and Lingfeng Wang. High-order graph convolutional network for skeleton-based human action recognition. In *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, pages 14–25. Springer, 2019.

[4] 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.

[5] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[6] 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.

[7] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7035–7043, 2017.

[8] Tianshui Chen, Muxin Xu, Xiaolu Hui, Hefeng Wu, and Liang Lin. Learning semantic-specific graph representation for multi-label image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 522–531, 2019.

[9] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1263–1272. JMLR. org, 2017.

[15] 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.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[17] Catalin Ionescu, Fuxin Li, and Cristian Sminchisescu. Latent structured models for human pose estimation. In *2011 International Conference on Computer Vision*, pages 2220–2227. IEEE, 2011.

[18] 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.

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

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

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

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[23] Y LeCun, Y Bengio, and G Hinton. Deep learning. nature 521 (7553), 436-444. *Google Scholar Google Scholar Cross Ref Cross Ref*, 2015.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] Francesc Moreno-Noguer. 3d human pose estimation from a single image via distance matrix regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2823–2832, 2017.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] Grégory Rogez, Jonathan Rihan, Srikumar Ramalingam, Carlos Orrite, and Philip HS Torr. Randomized trees for human pose detection. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] Wei Tang and Ying Wu. Does learning specific features for related parts help human pose estimation? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1107–1116, 2019.

[44] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 190–206, 2018.

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

[46] 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.

[47] 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.

[48] 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.

[49] Hashim Yasin, Umar Iqbal, Bjorn Kruger, Andreas Weber, and Juergen Gall. A dual-source approach for 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4948–4956, 2016.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.