Hierarchical Graph Networks for 3D Human Pose Estimation

Han Li[1]
qingshi9974@sjtu.edu.cn
Bowen Shi[1]
sjtu_shibowen@sjtu.edu.cn
Wenrui Dai*[1]
daiwenrui@sjtu.edu.cn
Yabo Chen[1]
chenyabo@sjtu.edu.cn
Botao Wang[2]
botaow@qti.qualcomm.com
Yu Sun[2]
sunyu@qti.qualcomm.com
Min Guo[2]
mguo@qti.qualcomm.com
Chenlin Li[1]
lcl1985@sjtu.edu.cn
Junni Zou[1]
zoujunni@sjtu.edu.cn
Hongkai Xiong[1]
xionghongkai@sjtu.edu.cn

[1] Shanghai Jiao Tong University
Shanghai, China

[2] Qualcomm AI Research
Shanghai, China

Abstract

Recent 2D-to-3D human pose estimation works tend to utilize the graph structure formed by the topology of the human skeleton. However, we argue that this skeletal topology is too sparse to reflect the body structure and suffer from serious 2D-to-3D ambiguity problem. To overcome these weaknesses, we propose a novel graph convolution network architecture, Hierarchical Graph Networks (HGN). It is based on denser graph topology generated by our multi-scale graph structure building strategy, thus providing more delicate geometric information. The proposed architecture contains three sparse-to-fine representation subnetworks organized in parallel, in which multi-scale graph-structured features are processed and exchange information through a novel feature fusion strategy, leading to rich hierarchical representations. We also introduce a 3D coarse mesh constraint to further boost detail-related feature learning. Extensive experiments demonstrate that our HGN achieves the state-of-the-art performance with reduced network parameters.

1 Introduction

3D human pose estimation aims to predict the 3D spatial coordinates of body joints from a monocular image and has been widely exploited in various applications such as abnormal behavior detection, sports analysis and automated driving. In recent years, 2D human pose estimation performance has been greatly improved owing to more refined network structure design and richer 2D human pose datasets. Recent works show that using such detected 2D joints positions, the 3D human pose can also be efficiently and accurately regressed [6, 10, 23, 29, 36]. To further boost performance, many attempts have been made to explicitly utilize the human skeletal topology and use graph convolutional networks (GCNs) to exploit the spatial configurations for 3D human pose estimation [4, 21, 37, 41]. However, the graph topology of human skeleton is usually formed by few number of joints (*e.g.* 17 joints in Human3.6M [16] dataset) and is sparse. In this paper, we raise a critical issue: are such few number of joints enough for reflecting the body structure and estimating 3D human pose?

We analyze the drawbacks of sparse graph representation from two perspectives. First, regressing the 3D human pose based on 2D joints positions is an ill-posed problem since multiple valid 3D poses can be projected to the same 2D pose. This inherent ambiguity problem will be more serious when the defined human skeleton structure is oversimple, thus hindering performance improvement. Second, each part of the human body is sparsely represented by only one joint, which will impede the expression of local information and lead to positioning failure when facing complex motions and occluded scenes.

In this paper, we address these issues by exploiting denser graph topology, proposing a novel architecture named *Hierarchical Graph Networks (HGN)*. Specifically, we propose a novel graph structure building strategy that utilizes the human shape information to obtain finer human structure representation. Then, starting from a sparse representation subnetwork, we gradually add sparse-to-fine representation subnetworks, and connect the multi-scale subnetworks in parallel. Since the mapping relation between sparse and fine representations is hard to model by manually designed graph pooling and unpooling, we also propose a multi-scale feature fusion strategy to learn the suitable mapping and exchange information across the parallel subnetworks . In Figure 1, we illustrate several typical GCN architectures for 2D-to-3D human pose estimation. The advantage of our HGN is that more delicate features are able to be extracted from the sparse-to-fine human structures.

Though promising it is, merely increasing the complexity of the graph structure without giving meaning to its nodes can only bring a limited performance improvement. Inspired by recent dense mesh vertices estimation methods [7], we conduct dense vertices coarsening to obtain the pseudo-groundtruth of the coarse mesh vertices and utilize it as an additional constraint. Since sparse mesh can represent the shape information of human body, it contains more abundant and detailed information. Adding additional mesh constraints makes our model extract detail-related features, which is of great help to the evaluation of some joints with high degrees of freedom.

In a nutshell, this paper makes the following contributions:

- We build a novel hierarchical graph network with multi-scale feature fusion. It is based on denser graph topology generated by our multi-scale graph structure building strategy and contains more delicate geometric information.
- We generate coarse mesh vertices pseudo-groundtruth by dense vertices coarsening and utilize it as an additional constraint to make the model pay more attention to detailed information.

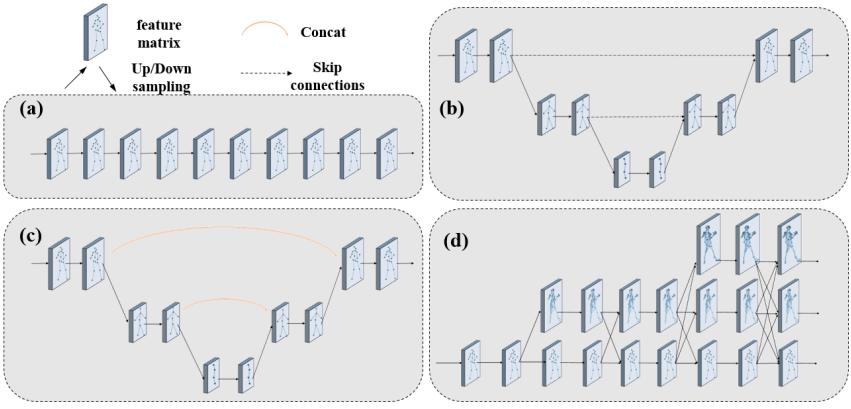

Figure 1: Several typical GCN architectures for 2D-to-3D human pose estimation: (a) Straight-forward architecture [41]. (b) Graph Stacked Hourglass [37] (c) Graph U-Nets [11]. (d) Ours Hierarchical Graph Networks. (b) and (c) also leverage multi-scale features but their largest scale graph is formed by human skeleton topology, while our architecture introduces denser graph topology and organizes the network in a sparse-to-fine way. Note that our architecture may look like HR-Net [31] but the network organization mode and feature fusion strategy are totally different.

- Experiment results demonstrate the superior performance of our HGN compared to other state-of-art GCN-based methods. The key idea of designing more delicate human structure representation may shed light on future research direction.

2 Related Work

3D Human pose estimation. Current 3D human pose estimation can be categorized into two types: one-stage approach and two-stage approach. The one-stage approach takes RGB images as input for 3D pose estimation. With the rapid development of deep learning, recent works [20, 25, 28, 30, 32, 43] take advantages of Convolutional Neural Networks (CNNs) for the image-to-3D human pose estiamtion. Zhou *et al*. [43] propose a weakly-supervised transfer learning method to make full use of mixed 2D and 3D labels, which augments the 2D pose estimation sub-network with a 3D depth regression sub-network to regress the depth. Sun *et al*. [33] employ soft-argmax operation to regress the 3D coordinates of body joints in a differentiable way. Pavlakos *et al*. [28] exploit voxel to discretize representations of the space around the human body and use 3D heatmaps to estimate 3D human pose.

The second category of approaches breaks the problem down into two steps: first predicting 2D human joints from the input image, and then lifting 2D joints to predict 3D pose. Our approach falls into this category. Martinez *et al*. [23] propose a simple yet effective baseline for 3D human pose estimation, it uses only 2D joints as input but gets highly accurate results, showing the importance of 2D joints information for 3D human pose estimation. Since the skeleton's topology can be viewed as a graph structure, there has been increasing use of Graph Convolutional Networks (GCNs) for 2D-to-3D pose estimation tasks [4, 19, 21, 37, 40, 41].

Graph Convolutional Networks. In recent years, Graph Convolutional Networks (GCNs) have been widely used to process graph-structured data, it can be regarded as a generaliza-

tion of traditional CNNs. In general, GCNs can be divided into two categories: spectral GCN [3, 9] and spatial GCN [13, 18, 27, 34], in which our approach falls into the second category. In the early day, Kipf and Welling [18] introduce the "vanilla" GCN, which performs the transformation and aggregation of graph-structured data via a simple graph convolution. Based on "vanilla" GConv, Zhao *et al*. [41] propose Semantic Graph Convolution (SemGConv), which can learn local and global semantic relations among nodes in the graph by adding a parameter matrix. Zou *et al*. [44] exploit a high-order GCN to learn long-range dependencies among body joints. However, they all adopt a straight-forward network architecture which simply stacks residual graph convolution blocks with same graph topology. To extract multi-scale features, Cai *et al*. [4] designed a U-nets like graph networks architecture and Xu *et al*. [37] proposed Graph Stacked Hourglass Networks. Unlike the above methods, we argue that the human skeletal graph used in these works is too sparse and propose hierarchical graph networks by exploiting denser graph topology.

3 Method

3.1 Preliminaries

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 edges. $\mathbf{A} \in \{0,1\}^{N\times N}$ is the adjacency matrix of $\mathcal{G}$, and we have $a_{ii} = 1$ and $a_{ij} = 1$ if j is the neighboring node of i. Each node i is associated with a D-dimensional feature vector $x_i \in \mathcal{R}^D$, and the node representations are collected into a matrix $\mathbf{X} \in \mathcal{R}^{D\times N}$. The vanilla GCN [18] is

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

where $\mathbf{X}' \in \mathcal{R}^{D'\times N}$ is the updated feature matrix and $\sigma(\cdot)$ is a non-linear function. $\tilde{\mathbf{A}}$ denotes a symmetrically normalized version of $\mathbf{A}$ and $\mathbf{W} \in \mathcal{R}^{D'\times D}$ is a learnable matrix that transforms node representations. SemGConv [41] further learns the semantic relationships of neighboring nodes by adding another learnable weighting matrix $\mathbf{T} \in \mathcal{R}^{N\times N}$.

$$\mathbf{X}' = \sigma\left(\mathbf{W}\mathbf{X}\rho_i\left(\mathbf{T}\odot\tilde{\mathbf{A}}\right)\right), \tag{2}$$

where $\odot$ is an element-wise product operation and $\rho_i(\cdot)$ is softmax nonlinearity which normalizes the input matrix across all choices of i. Following previous works [21, 38, 41], we use two different transformation matrices for the representation of each node i and its neighbors respectively in actual implementation.

3.2 Hierarchical Graph Networks

As mentioned in Section 1, the 2D-to-3D human pose estimation is an ill-posed problem, and it will become more serious when occlusion occurs. We believe that the above problem can be alleviated if we leverage denser graph structures to depict the human skeleton, thus increasing the complexity of the constraint relationship between human joints. To achieve this goal, as shown in Figure 2, we propose Hierarchical Graph Networks (HGN) consisting of three subnetworks with different scales of graph structure. Novel graph structure building strategy and multi-scale feature fusion strategy are also designed for HGN.

Multi-scale graph structure building. We build three different human structure graphs and assign them to the three subnetworks of HGN in a sparse-to-fine way. The sparsest graph for the bottom subnetwork in Figure 2 is defined on the standard human skeletal graph $\mathcal{G}_P = \{\mathcal{V}_P, \mathcal{E}_P\}$. Similar to existing methods [21, 23, 44], the number of nodes J is set to 17 in this paper. For the middle and top subnetworks, we coarsen the human mesh graph

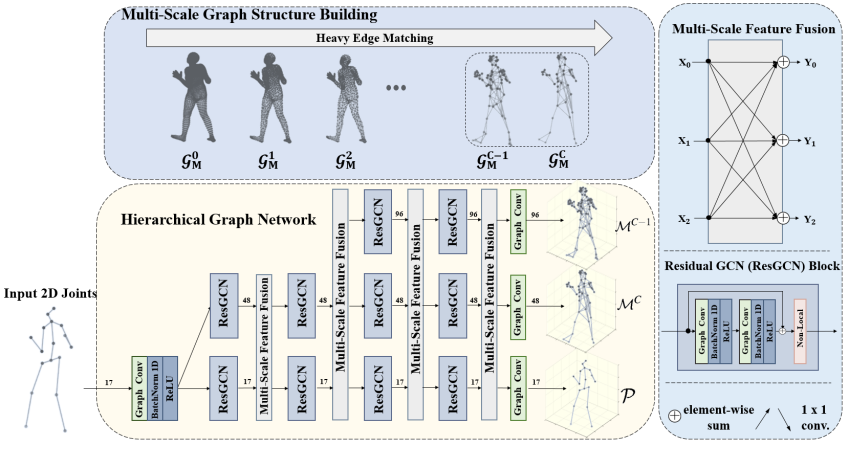

Figure 2: The overall pipeline of HGN. The input 2D joints $\mathbf{J}$ passes three subnetworks of HGN to obtain multi-scale features and output the predicted 3D Pose $\mathcal{P}$ and coarse mesh $\mathcal{M}^C$ and $\mathcal{M}^{C-1}$ in parallel. The nodes of graph built for three subnetworks from bottom to up is 17, 48 and 96, respectively, using the multi-scale graph structure building processing shown in the upper part. The down-right part illustrates the Residual GCN (ResGCN) block and the up-right part shows the proposed multi-scale feature fusion strategy used in HGN.

$\mathcal{G}_M = \{\mathcal{V}_M, \mathcal{E}_M\}$, which represents human shape information and contains 6890 vertices, multiple times using Heavy Edge Matching (HEM) [9] and obtain a set of various scales of graphs $\{\mathcal{G}_M^c\}_{c=0}^C$, where c indicates the coarsening level. Although more complex graphs can be allocated, we choose the two coarsest graphs $\mathcal{G}_M^{C-1}$ and $\mathcal{G}_M^C$ whose the number of nodes V^c equal to 96 and 48, respectively, to maintain a suitable model size.

Multi-scale feature fusion. Since the human skeletal graph $\mathcal{G}_P$ and the coarse human mesh graphs $\mathcal{G}_M^{C-1}$ and $\mathcal{G}_M^C$ are constructed in different ways, it is difficult to describe the specific corresponding relationship between them in a manual way or simply graph downsampling and upsampling operations [4, 37]. As an alternative, we design a multi-scale feature fusion strategy to learn their mapping relations.

Inspired by the exchange units proposed by Sun *et al.* [31], we conduct multi-scale fusions such that the parallel subnetworks can exchange different scales information from each other. Given the input feature with different number of nodes $\{\mathbf{X_0}, \mathbf{X_1}, \ldots, \mathbf{X_s}\}$, where the subscript denotes the graph scale, we can obtain the fused feature $\{\mathbf{Y_0}, \mathbf{Y_1}, \ldots, \mathbf{Y_s}\}$ whose scale and widths are the same to the input by

$$\mathbf{Y}_k = \sum_{i=1}^{s} \mathrm{a}(\mathbf{X}_i, k), \ \ k = 0, 1, \ldots, s, \tag{3}$$

where function $\mathrm{a}(\mathbf{X}_i, k)$ consists of upsampling or downsampling $\mathbf{X}_i$ from graph scale i to scale k. We use 1×1 convolution for both upsampling and downsampling since the mapping relations are proved to be learned well in such a simple way.

Network architecture. As shown in Figure 2, the input 2D pose is mapped to the latent space by a pre-processing graph convolution layer. Then, the multi-scale features extracted in the three subnetworks are fused repeatedly. The subnetworks have 4, 4, 2 residual blocks [14], respectively from bottom to up. Each residual block consists of two graph convolution layers and is followed by a non-local layer [35] to capture both local and global information, and all the graph convolution layers are followed by batch normalization [15]

and a ReLU non-linear layer [26]. Finally, the features are fed into the output convolution layer and mapped to the output space. It is noted that our HGN does not rely on a specific graph convolution method, so both SemGConv and Vanilla GConv introduced in Section 3.1 can be implemented in our architecture.

3.3 Mesh Constraint

Since graphs $\mathcal{G}_M^{C-1}$ and $\mathcal{G}_M^C$ are derived from mesh graph $\mathcal{G}_M$, we can also generate coarse mesh vertices pseudo-groundtruth by dense vertices coarsening and leverage it as another constraint to further refine the local feature representation. Specifically, we generate 3D human mesh $\mathbf{M}$ with groundtruth vertices location by fitting SMPL parameters to the 3D groundtruth poses using SMPLify-X [2], and then obtain the pseudo-groundtruth of two coarsest meshes $\mathbf{M}^{C-1} \in \mathcal{R}^{V_{C-1}\times 3}$ and $\mathbf{M}^C \in \mathcal{R}^{V_C\times 3}$ by a pre-defined indices mapping operation. The final loss function is a combination of 3D pose and 3D coarse mesh constraint:

$$\mathcal{L}(\mathcal{P},\mathcal{M}^{C-1},\mathcal{M}^C) = \underbrace{\lambda_P \sum_{i=1}^{J} \|\hat{\mathbf{p}}_i - \mathbf{p}_i\|_2^2}_{\text{3D pose loss}} + \underbrace{\lambda_M \left(\sum_{i=1}^{V_{C-1}} \|\hat{\mathbf{m}}_i^{C-1} - \mathbf{m}_i^{C-1}\|_2^2 + \sum_{i=1}^{V_C} \|\hat{\mathbf{m}}_i^{C} - \mathbf{m}_i^{C}\|_2^2 \right)}_{\text{3D coarse mesh loss}}, \tag{4}$$

where $\lambda_P = 1$, $\lambda_M = 0.01$. $\mathcal{P} = \{\hat{\mathbf{p}}_i | i = 1,\ldots,J\}$ are the predicted 3D pose and $\mathcal{M}^c = \{\hat{\mathbf{m}}_i^{\mathbf{c}} | i = 1,\ldots,V_c,\ c = C-1, C\}$ are the predicted 3D coarse meshes. $\mathbf{p}_i$ and $\mathbf{m}_i^c$ are the groundtruth /pesudo-groudtruth corresponding to $\hat{\mathbf{p}}_i$ and $\hat{\mathbf{m}}^{\mathbf{c}}$.

4 Experiments

4.1 Experimental Setups

Dataset. We mainly evaluate our proposed method on the Human3.6M dataset [16], which is widely used in the 3D human pose estimation task. It provides 3.6 million color images taken from four synchronous cameras in different positions and perspectives, by recording 11 subjects actors performing 15 different actions, such as eating and walking. There are 7 subjects annotated with 3D joints. For fair comparison, we follow previous works [23, 37, 41] and choose 5 subjects (S1, S5, S6, S7, S8) for training and the other 2 subjects (S9, S11) for test. Besides, the MPI-INF-3DHP [24] test set provides images in three different scenarios: studio with a green screen (GS), studio without green screen (noGS) and outdoor scene (Outdoor). We apply our model to this dataset to test the generalization capabilities of our proposed method.

Evaluation. For the Human3.6M benchmark, there are two evaluation protocols used in previous works [23, 37, 41]. Protocol #1 uses the mean per-joint position error (MPJPE) as evaluation metric, which computes the mean Euclidean distance error per-joints between the predicted 3D joints and the ground truth after the origin (pelvis) alignment. Protocol #2 aligns the predicted 3D joints with the ground truth by rigid transformation and then computes the error. This metric is abbreviated as PA-MPJPE. Both of these two metrics are measured in millimeter (mm).

Implement details. We implement our method within the PyTorch framework. During the training stage, we choose the Adam optimizer [17] with the learning rate initialized to 0.001 and decayed by 0.9 per 20 epochs. We train each model for 100 epochs using a mini-batch size of 64. We initialize weights of the GCNs using the initialization method described

Table 1: Ablation study on effects of mesh constraint. Different weights of mesh constraint λ_M in Equation (4) are set and $\lambda_M = 0$ means removing the mesh constraint.

λ_P	λ_M	MPJPE (mm)	PA-MPJPE (mm)
1	0	38.31	29.04
1	0.001	37.68	28.79
1	0.01	**37.32**	**28.51**
1	0.1	38.69	29.09
1	1	39.64	29.81

Table 2: Ablation study on effects of denser graph topology with λ_M set to 0.01.

Method	MPJPE (mm)	PA-MPJPE (mm)	MPVPE		# Params
			$\mathcal{M}^{C-1}$	$\mathcal{M}^{C}$	
SemGCN (Baseline) [41]	40.78	31.01	-	-	0.43M
HGN w/o $\mathcal{G}_M^C$	38.69	29.07	66.28	-	0.82M
HGN w/o $\mathcal{G}_M^{C-1}$	37.83	28.71	-	65.92	0.81M
HGN	**37.32**	**28.51**	62.74	61.76	1.04M

in [12]. To avoid overfitting, we also adopt Max-norm regularization. In the following experiment, unless specified, the SemGConv is used as the graph convolution layer.

4.2 Ablation Study

We conduct a series of ablation studies to better understand how each component affects the performance. The 2D ground truth is taken as input.

Effects of mesh constraint. We first diagnose how the mesh constraint affects the performance. As shown in Table 1, our method achieves the best performance when setting the weight of coarse mesh constraint λ_M to 0.01. Compared with not adding mesh constraint ($\lambda_M = 0$), we achieve 0.99mm (38.31 to 37.32) and 0.53mm (29.04 to 28.51) gain under two protocols respectively. The performance gain is mainly due to that the mesh constraint enriches the fine-grained representation. However, if we increase λ_M, the performance dramatically drops. We believe that this is because the pose-related information will be covered by shape-related information using higher mesh constraint weight. Therefore, it makes sense to set a small weight for mesh constraint for better pose estimation.

Effects of denser graph topology. We then inspect how denser graph topology benefits the representation.We first set λ_M to 0.01 and carry out experiments with three variants of our HGN. 1) **SemGCN (Baseline)**: only the bottom subnetwork in Figure 2 defined on $\mathcal{G}_P$ is reserved. This straight-forward architecture is equivalent to Semantic Graph Convolutional Networks (SemGCN) [41], and we treat it as our baseline. 2) **HGN w/o $\mathcal{G}_M^C$**: we remove the top subnetwork and change the graph structure in the middle subnetwork from $\mathcal{G}_M^C$ to $\mathcal{G}_M^{C-1}$, so that the network can output 3D pose and coarse mesh $\mathcal{M}^{C-1}$ containing 96 vertices in parallel. 3) **HGN w/o $\mathcal{G}_M^{C-1}$**: we remove the top subnetwork and do not modify the other subnetworks to output 3D pose and coarse mesh $\mathcal{M}^C$ containing 48 vertices. The results are shown in Table 2, in which the mean per vertex position error (MPVPE) for mesh prediction measurement is also listed as a reference. We find that all networks with delicate graph structures outperform the baseline for a large margin, which proves the benefits of introducing denser graph topology. Our HGN achieves the best results in MPJPE (37.32mm), indicating that our proposed model has a strong ability to leverage sparse-to-fine graph structures.

Furthermore, we set λ_M to 0 to remove the influence of mesh constraint and model parameters. Experiments are made by fixing the number of channels and model parameters,

Table 3: Ablation study on the effect of denser graph topology with λ_M set to 0. The number of channels and model parameters are fixed (to 128 and 0.43M) for evaluations, respectively.

Method	# Channels	MPJPE (mm)	PA-MPJPE (mm)	# Params
SemGCN (Baseline) [41]	128	40.78	31.01	0.43M
HGN w/o $\mathcal{G}_M^C$	128	38.94	29.47	0.82M
HGN w/o $\mathcal{G}_M^{C-1}$	128	38.59	29.18	0.81M
HGN	128	**38.31**	**29.04**	1.04M
HGN w/o $\mathcal{G}_M^C$	91	39.87	30.31	0.43M
HGN w/o $\mathcal{G}_M^{C-1}$	92	39.71	30.32	0.43M
HGN	79	**39.26**	**29.60**	0.43M

Table 4: Comparison of GraphSH [37] and our method using SemGConv and vanilla GConv.

(a) SemGConv

Method	Channels	MPJPE (mm)	# Params
SeqRes [41]	128	40.78	0.43M
GraphSH [37]	64	39.20	0.44M
Ours	64	38.74	0.29M
Ours	128	**37.32**	1.04M

(b) Vanilla GConv

Method	Channels	MPJPE (mm)	# Params
SeqRes [41]	128	65.90	0.30M
GraphSH [37]	64	59.10	0.22M
Ours	64	42.92	0.21M
Ours	128	**40.66**	0.71M

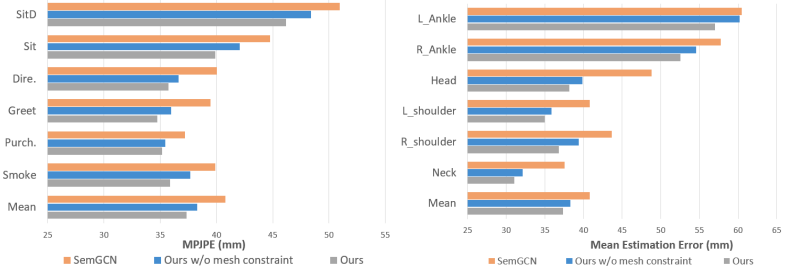

Figure 3: Mean estimation errors on different body parts (left) and actions (right). Ground truth 2D keypoints are used as input. We show the result for [41] and ours.

respectively. Table 3 shows that our HGN still achieves the overall best performance.

Effects of hierarchical network structures. To verify that our architecture has better performance than some typical structures, we keep the GCN type same and the model size comparable for fair comparison. Table 4 shows that the model using our architecture performs better than Sequential Residual blocks (denoted as SeqRes) [41] and Graph Stacked Hourglass (denoted as GraphSH) [37] architecture with fewer parameters. It is noted that our method boosts the performance for a large margin when using traditional Vanilla GConv, which demonstrates the great advantage of our architecture itself.

Understanding the performance improvement. We present the average estimation errors of different body parts and actions as well as the overall mean errors in Figure 3. Among all the actions, our method obtains larger gains for those with serious self-occlusion, such as *'Sitting'* (4.78mm), *'SittingDown'* (4.85mm), *'Greeting'* (4.74mm), etc, while the overall gain is 3.16mm. For body parts, our method brings much improvement for *'head'* (10.64mm), *'right shoulder'* (6.88mm) and *'left shoulder'* (5.83mm) etc because the vertices of coarse mesh, as shown in Figure 2, are denser in the upper part of the human body. Those results prove our opinion that more complex graph structure can bring benefit to the depiction of the human skeleton and some joints with high degrees of freedom.

Table 5: Quantitative evaluation results using MPJPE in millimeter on Human3.6M under Protocol #1. No rigid alignment or transform is applied in post-processing. **Detected 2D keypoints** are used as input. † uses temporal information. + uses extra data from MPII dataset. Best results are highlighted in bold.

Protocol #1	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Puch.	Sit	SitD.	Smoke	Wait	WalkD	Walk	WalkT	**Avg.**
Pavllo *et al.* [29]†	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
Cai *et al.* [4]†	44.6	47.4	45.6	48.8	50.8	59.0	49.7	47.2	43.9	57.9	61.9	46.6	51.3	37.1	39.4	48.8
Martinez *et al.* [23]	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
Pavlakos *et al.* [28]+	48.5	54.4	54.5	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.* [40]	48.2	60.8	51.8	64.0	64.6	**53.6**	51.1	67.4	88.7	**57.7**	73.2	65.6	**48.9**	64.8	51.9	60.8
Ci *et al.* [8]+	46.8	52.3	**44.7**	**50.4**	**52.9**	68.9	49.6	46.4	60.2	78.9	**51.2**	50.0	54.8	40.4	43.3	52.7
Liu *et al.* [21]	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
Zou *et al.* [42]	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
Xu & Takano [37]	**45.2**	**49.9**	47.5	50.9	54.9	66.1	48.5	46.3	**59.7**	71.5	51.4	**48.6**	53.9	**39.9**	44.1	51.9
Ours	47.8	52.5	47.7	50.5	53.9	60.7	49.5	49.4	60.0	66.3	51.8	48.8	55.2	40.5	**42.6**	**51.8**

Table 6: Quantitative evaluation results using PA-MPJPE in millimeter on Human3.6M under Protocol #2. Rigid alignment is applied in post-processing. **Detected 2D keypoints** are used as input. † uses temporal information. + uses extra data from MPII dataset. Best results are highlighted in bold.

Protocol #2	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD	Walk	WalkT	**Avg.**
Pavllo *et al.* [29]†	34.2	36.8	33.9	37.5	37.1	43.2	34.4	33.5	45.3	52.7	37.7	34.1	38.0	25.8	27.7	36.8
Cai *et al.* [4]†	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
Martinez *et al.* [23]	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
Pavlakos *et al.* [28]+	**34.7**	39.8	41.8	**38.6**	42.5	47.5	38.0	36.6	50.7	56.8	42.6	39.6	43.9	32.1	36.5	41.8
Ci *et al.* [8]+	36.9	41.6	38.0	41.0	41.9	51.1	38.2	37.6	49.1	62.1	43.1	39.9	43.5	32.2	37.0	42.2
Liu *et al.* [21]	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
Zou *et al.* [42]	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
Ours	35.8	**39.7**	**36.3**	40.6	**40.2**	**45.9**	**36.8**	**35.8**	**47.3**	**53.7**	**40.7**	**36.4**	43.1	**29.8**	**32.8**	**39.6**

Table 7: Results on the test set of MPI-INF-3DHP [24] by scene. The results are shown in PCK and AUC.

	Trainning data	GS	noGS	Outdoor	ALL (PCK ↑)	ALL (AUC ↑)
Martinez *et al.* [23]	H36M	49.8	42.5	31.2	42.5	17.0
Mehta *et al.* [24]	H36M	70.8	62.3	58.8	64.7	31.7
Yang *et al.* [39]	H36M+MPII	-	-	-	69.0	32.0
Zhou *et al.* [43]	H36M+MPII	71.1	64.7	72.7	69.2	32.5
Luo *et al.* [22]	H36M	71.3	59.4	65.7	65.6	33.2
Ci *et al.* [8]	H36M	74.8	70.8	77.3	74.0	36.7
Zhou *et al.* [42]	H36M+MPII	75.6	71.3	80.3	75.3	38.0
Xu and Takano [37]	H36M	81.5	81.7	75.2	80.1	45.8
Ours	H36M	**87.0**	**84.9**	**82.7**	**85.2**	**52.1**

4.3 Comparison With the State-of-the-Art

We use the cascaded pyramid network (CPN) [5] as 2D pose detector to obtain 2D input joints for benchmark evaluation. CPN is pre-trained on COCO-dataset and fine-tuned on Human3.6M. Following previous works [4, 29, 43], we also perform horizontal flip augmentation. The results are shown in Tables 5 and 6 for the two protocols, respectively. Note that some other methods [4, 29] that focus on video-based 3D pose estimation are complementary to our method and can be used to improve the performance.

Tables 5 and 6 show that the performance improvement of our model is significant, outperforming all other GCN-based methods and some representative methods [8, 28] using

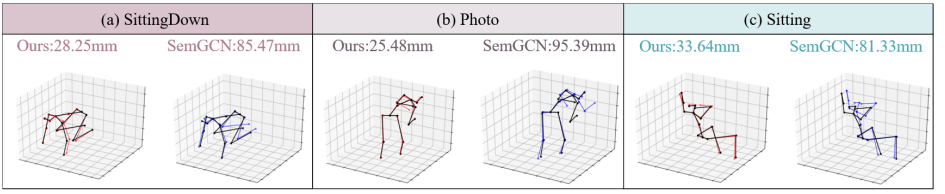

Figure 4: Qualitative results of actions with self-occlusion, i.e., SittingDown, Photo, and Sitting. 3D groud truth, ours, and SemGCN are shown in black, red, and blue, respectively.

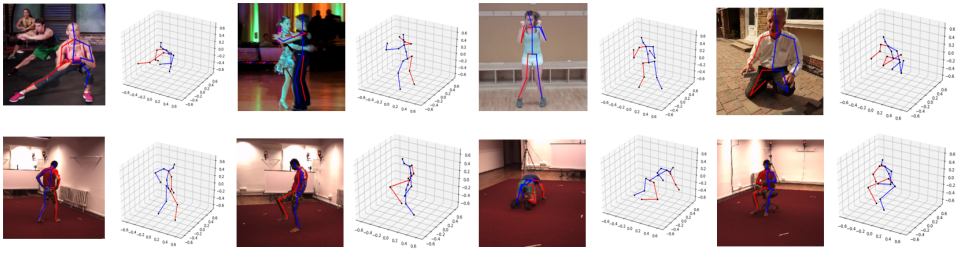

Figure 5: Qualitative results of our method on Human3.6M [16] (bottom) and MPII [1] (top).

extra MPII data [1]. GraphSH [37] achieves comparable performance, but it uses more parameters (3.70M) while our model has only 1.04M parameters, and we have proved in Section 4.2 that our method surpasses GraphSH a lot with the same GCN type and channels. The results demonstrate the great advantage of our HGN.

Figure 4 shows qualitative results for those actions with serious self-occlusion. Compared with baseline, our HGN can alleviate the depth ambiguity caused by self-occlusion. Figure 5 demonstrates more qualitative results of our HGN on the Human3.6M and MPII datasets. Here, MPII contains in-the-wild images that are unseen for the model trained on Human3.6M. These results further validate the strong generalization ability of our method.

4.4 Generalization Ability

The MPI-INF-3DHP test set [24] provides images in three different scenarios: studio with a green screen (GS), studio without green screen (noGS) and outdoor scene (Outdoor). We apply our model to this dataset to test the generalization capabilities of our proposed method and employ 3D-PCK and AUC as evaluation metrics. As shown in Tab. 7, our model yields 85.2 in PCK and 52.1 in AUC while only using the Human3.6M dataset for training, which outperforms all the previous state-of-the-arts. These results validate the strong generalization capability of our architecture.

5 Conclusion

In this paper, we propose a novel architecture named *Hierarchical Graph Networks (HGN)*. The main contributions are two folds. First, we build a novel sparse-to-fine architecture with multi-scale feature fusion based on the denser graphs generated by a multi-scale graph structure building strategy for better feature extraction. Second, we leverage the human coarse mesh as an additional constraint, refining the local feature representation. Extensive experiment results reveal the benefit of our design.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61971285, Grant 61871267, Grant 61972256, Grant 61720106001, Grant 61931023, Grant 61932022, Grant 61831018, and in part by the Program of Shanghai Science and Technology Innovation Project under Grant 20511100100.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, 2014.

[2] 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 *Proceedings of the European conference on computer vision*, pages 561–578, 2016.

[3] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[4] 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 *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2272–2281, 2019.

[5] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7103–7112, 2018.

[6] Yu Cheng, Bo Yang, Bo Wang, Wending Yan, and Robby T Tan. Occlusion-aware networks for 3d human pose estimation in video. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 723–732, 2019.

[7] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2Mesh: Graph convolutional network for 3D human pose and mesh recovery from a 2D human pose. In *Proceedings of the European Conference on Computer Vision*, pages 769–787, 2020.

[8] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3D human pose estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2262–2271, 2019.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems 29*, pages 3844–3852, 2016.

[10] 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 *32nd AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 6821–6828, 2018.

[11] Hongyang Gao and Shuiwang Ji. Graph U-Nets. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2083–2092, 2019.

[12] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010*, pages 249–256, 2010.

[13] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 31*, page 1025–1035, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456, 2015.

[16] 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.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *2nd International Conference on Learning Representations*, 2014.

[18] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations*, 2016.

[19] Deying Kong, Haoyu Ma, and Xiaohui Xie. SIA-GCN: A spatial information aware graph neural network with 2D convolutions for hand pose estimation. In *British Machine Vision Conference*, 2020.

[20] Jiahao Lin and Gim Hee Lee. HDNet: Human depth estimation for multi-person camera-space localization. In *Proceedings of the European Conference on Computer Vision*, pages 633–648, 2020.

[21] 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*, pages 318–334, 2020.

[22] Chenxu Luo, Xiao Chu, and Alan Yuille. Orinet: A fully convolutional network for 3d human pose estimation. In *British Machine Vision Conference*, 2020.

[23] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3D human pose estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2640–2649, 2017.

[24] 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 *international conference on 3D vision*, pages 506–516. IEEE, 2017.

[25] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Camera distance-aware top-down approach for 3D multi-person pose estimation from a single RGB image. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 10133–10142, 2019.

[26] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814, 2010.

[27] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 2014–2023, 2016.

[28] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3D human pose estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7307–7316, 2018.

[29] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3D human pose estimation in video with temporal convolutions and semi-supervised training. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7753–7762, 2019.

[30] Bowen Shi, Yuhui Xu, Wenrui Dai, Botao Wang, Shuai Zhang, Chenglin Li, Junni Zou, and Hongkai Xiong. Tiny-Hourglassnet: An efficient design for 3D human pose estimation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1491–1495, 2020.

[31] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5693–5703, 2019.

[32] Xiao Sun, Jiaxiang Shang, Shuang Liang, and Yichen Wei. Compositional human pose regression. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2602–2611, 2017.

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

[34] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *6th International Conference on Learning Representations*, 2018.

[35] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

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

[37] Tianhan Xu and Wataru Takano. Graph stacked hourglass networks for 3D human pose estimation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16105–16114, 2021.

[38] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *32nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 7444–7452, 2018.

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

[40] Ailing Zeng, Xiao Sun, Lei Yang, Nanxuan Zhao, Minhao Liu, and Qiang Xu. Learning skeletal graph neural networks for hard 3d pose estimation. *arXiv preprint arXiv:2108.07181*, 2021.

[41] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3D human pose regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3425–3435, 2019.

[42] Kun Zhou, Xiaoguang Han, Nianjuan Jiang, Kui Jia, and Jiangbo Lu. Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2344–2353, 2019.

[43] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3D human pose estimation in the wild: A weakly-supervised approach. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 398–407, 2017.

[44] Zhiming Zou, Kenkun Liu, Le Wang, and Wei Tang. High-order graph convolutional networks for 3D human pose estimation. In *British Machine Vision Conference*, 2020.