Learning Global Pose Features in Graph Convolutional Networks for 3D Human Pose Estimation

Kenkun Liu[1], Zhiming Zou[1], and Wei Tang[1]*

University of Illinois at Chicago, Chicago, IL, USA
{kliu44,zzou6,tangw}@uic.edu

Abstract. As the human body skeleton can be represented as a sparse graph, it is natural to exploit graph convolutional networks (GCNs) to model the articulated body structure for 3D human pose estimation (HPE). However, a vanilla graph convolutional layer, the building block of a GCN, only models the local relationships between each body joint and their neighbors on the skeleton graph. Some global attributes, e.g., the action of the person, can be critical to 3D HPE, especially in the case of occlusion or depth ambiguity. To address this issue, this paper introduces a new 3D HPE framework by learning global pose features in GCNs. Specifically, we add a global node to the graph and connect it to all the body joint nodes. On one hand, global features are updated by aggregating all body joint features to model the global attributes. On the other hand, the feature update of each body joint depends on not only their neighbors but also the global node. Furthermore, we propose a heterogeneous multi-task learning framework to learn the local and global features. While each local node regresses the 3D coordinate of the corresponding body joint, we force the global node to classify an action category or learn a low-dimensional pose embedding. Experimental results demonstrate the effectiveness of the proposed approach.

1 Introduction

The objective of 3D human pose estimation (HPE) is to predict the positions of human body joints in the camera coordinate system from a single RGB image. This task gains a lot of attention in the last few years [1–10] since it has various applications in human-computer interaction, action recognition and motion capture. 3D HPE is essentially an ill-posed problem because one pose in the 2D image coordinate system may correspond to multiple poses in the 3D camera coordinate system. But this ambiguity can be alleviated to a large extent by exploiting the structure information of the human body [11, 12].

Two streams of approaches for 3D HPE have been investigated. The first stream of methods aim to build an end-to-end system that predicts the 3D coordinates of body joints directly from the input image [3, 13]. Early approaches

* Corresponding author.

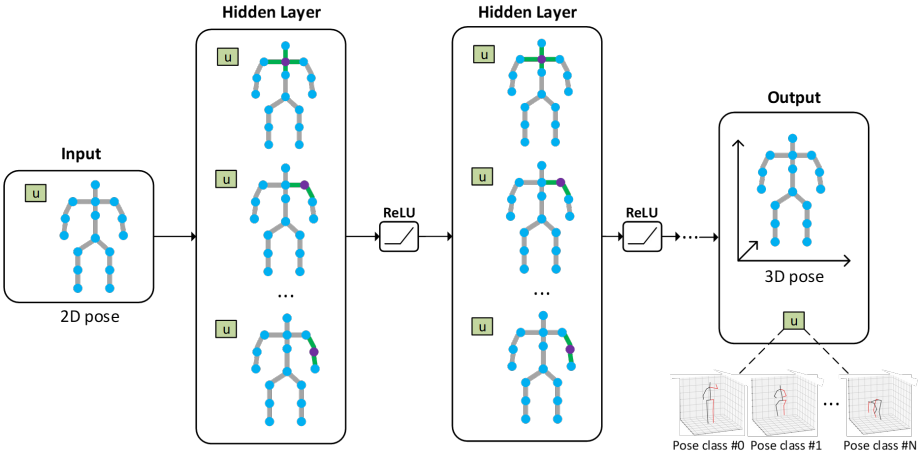

Fig. 1. Illustration of a graph convolutional network (GCN) with a global node u for 3D human pose estimation. The input is the 2D body joint locations predicted by an off-the-shelf 2D pose detector with a zero-initialized global feature vector. The GCN repeatedly transforms and aggregates features of local and global nodes to learn increasingly powerful representations. Finally, it predicts the 3D pose as well as the output of an auxiliary task, e.g., an action label of the pose.

[14, 15] are based on hand-designed features but they are likely to fail in some challenging scenarios, e.g., depth ambiguity, viewpoint variation and occlusion. In recent years, the development of convolutional neural networks pushes the edge of this problem and significantly improves the estimation accuracy with the help of large-scale image data [16–20]. The second stream of approaches divide the 3D HPE into two subtasks, i.e., the prediction of 2D joints locations and 2D-to-3D pose regression [1, 11, 12]. Martinez et al. [1] prove that 3D coordinates of human body joints could be accurately estimated merely from the output of a 2D pose detector.

To model the articulated body structure, graph convolutional networks (GCNs) [21, 22] have been introduced to solve the 2D-to-3D pose lifting problem [5, 23, 11]. GCNs are generalized from CNNs to construct a non-linear mapping in a graph domain. Different from CNNs, which act on image patches, GCNs update the features of each node from its neighbouring nodes in a graph. In this way, the prior of the graph structure is fed into the GCN model.

Though GCNs have shown decent results in 2D-to-3D pose lifting [11, 23, 5], they have one potential limitation. A vanilla graph convolutional layer, the building block of a GCN, only models the local relationships between each body joint and their neighbors. Some global attributes, e.g., the action or viewpoint of the person, can be critical to 3D HPE, especially in the case of occlusion or depth ambiguity. Unfortunately, the importance of global features to 3D HPE is largely ignored by prior work.

This paper introduces a new 3D HPE framework by learning global pose features in GCNs. Specifically, we first add a global node to the graph and connect it to all the body joint nodes. On one hand, the global node aggregates features from all body joints to model the global attributes. On the other hand, the feature update of each body joint depends on not only their neighbors but also the global node. To facilitate the learning of meaningful global attributes, we propose a heterogeneous multi-task learning framework. Specifically, we introduce auxiliary learning tasks for the global node. While each local node regresses the 3D coordinates of the corresponding body joint, we force the global node to classify an action category or learn a low-dimensional pose embedding.

Extensive ablation study indicates that (1) learning global features in a GCN can improve its performance and (2) solving the auxiliary learning tasks together with 3D HPE is beneficial.

In sum, the contribution of this paper is threefold.

- To our knowledge, this is the first work to learn global pose features in a GCN for 3D HPE. We add a global node to the skeleton graph and connect it to every body joint node so that each local node has access to global information during feature update.
- We propose a heterogeneous multi-task learning framework to facilitate the learning of effective global representations in a GCN. We introduce two auxiliary learning tasks, i.e., action classification and pose embedding, to achieve this goal.
- We perform extensive ablation study to investigate whether the extra global node and the auxiliary tasks help 3D HPE. Experimental results indicate that the proposed approach can outperform some state-of-the-art methods.

2 Related Work

3D Human Pose Estimation. The last two decades have seen the rapid development of 3D HPE. Early work builds 3D HPE systems on handcrafted features and geometric constraints [24–26]. Recently, state-of-the-art methods are based on deep neural networks. Chen et al. [27] propose a weekly-supervised encoder-decoder framework that can learn geometry-aware representations using only 2D annotations. Wang et al. [12] design a new network architecture to learn the bi-directional dependencies of body parts. 3D HPE can also be divided into two subtasks, i.e., 2D HPE and 2D-to-3D lifting. For example, Martinez et al. [1] use a fully connected network to regress the 3D body joint locations from the output of an off-the-shelf 2D pose detector. This simple baseline is very effective and outperforms the state-of-the-art one-stage approaches.

The works most related to ours are [11, 23, 5, 28, 29], which also apply GCNs for 3D pose regression. Zhao et al. [11] propose a semantic GCN to learn semantic information not explicitly represented in the graph. Ci et al. [23] extend the GCN to a locally connected network to improve its representation capability. Cai et al. [5] introduce a local-to-global network to learn multi-scale features for the graph-based representations. Liu et al. [28] study different weight sharing methods in

the graph convolution. Zou et al. [29] introduce a high-order GCN for 3D HPE. However, the main contribution of this paper is to learn global pose features in a GCN, which these prior approaches ignore. Furthermore, while they only focus on the task of 3D HPE, we introduce a heterogeneous multi-task learning framework with auxiliary tasks to facilitate the learning of global features. And we generate labels for global node supervision by ourselves. The global features in our setting can be directly used for other follow-on tasks.

Graph Convolutional Networks. GCNs generalize CNNs by performing convolutions on graph data. They have been widely used to solve problems like the citation network [21] and molecular property prediction [30]. There are two categories of GCNs: spectral approaches and non-spectral (spatial) approaches [22]. The former are defined in the Fourier domain by calculating the eigen-decomposition of graph Laplacian [31], while the latter apply neural message passing to features defined on a graph [30]. Our approach falls into the second category. Battaglia et al. [32] generalize previous work into a unified graph network and also discuss the use of global node. While they focus on graph or node classification, our model is specially designed for 3D HPE. More importantly, we introduce a heterogeneous multi-task learning framework to learn global pose features via auxiliary tasks.

3 Approach

In this section, we first revisit the vanilla GCN, which models the local relationship between each node and their neighbors. Then, we propose to learn global pose features in a GCN and introduce a heterogeneous multi-task learning framework. Finally, we discuss the network architecture for 3D HPE.

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. We can represent the collection of all edges via an adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$. Let $\mathbf{x}_i \in \mathcal{R}^D$ denote a D-dimensional feature vector corresponding to each node i. $\mathbf{X} \in \mathcal{R}^{D \times N}$ collects all feature vectors, whose i-th column is $\mathbf{x}_i$. Then a graph convolutional layer [21], the building block of a GCN, updates features defined on the nodes through the following operation:

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

where $\mathbf{X}' \in \mathbb{R}^{D' \times N}$ is the updated feature matrix, D' is the dimension of the updated feature vector of each node, $\sigma(\cdot)$ is an activation function, e.g., ReLU, $\mathbf{W} \in \mathbb{R}^{D' \times D}$ is a learnable weight matrix. $\hat{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\tilde{\mathbf{D}}^{-\frac{1}{2}}$ is a normalized version of $\mathbf{A}$. Adding an identity matrix $\mathbf{I}$ to $\mathbf{A}$ means to include self-connections in the graph so that the update of a node feature vector also depends on itself. $\tilde{\mathbf{D}}$ is the diagonal node degree matrix of $\mathbf{A} + \mathbf{I}$ and helps the graph convolution to retain the scale of features.

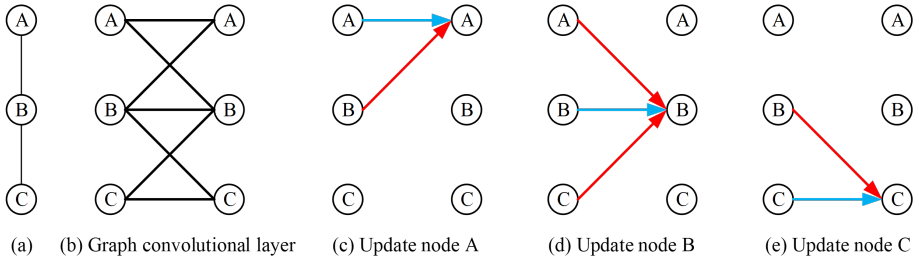

Fig. 2. Illustration of the feature update in a graph convolutional layer. Blue arrows and red arrows respectively correspond to self-connections and other-connections. (a) A simple graph consisting of three nodes. (b) The updated features of each node (the right side) depend on the input features of itself and its neighboring nodes (the left side). (c)(d)(e) respectively show the feature update of nodes A, B and C.

A GCN takes as input a feature vector associated with each node and repeatedly transforms them via a composition of multiple graph convolutions to get increasingly more powerful representations, which are used by the last layer to predict the output.

Let $\hat{a}_{ij}$ be the entry of $\hat{\mathbf{A}}$ at (i, j). $\mathcal{N}_i$ and $\hat{\mathcal{N}}_i \equiv \mathcal{N}_i \cup \{i\}$ denote the neighbors of node i excluding and including the node itself respectively. This means $j \in \hat{\mathcal{N}}_i$ if and only if $\hat{a}_{ij} \neq 0$. Then Eq. (1) can be written equivalently as below.

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

where $i \in \{1, ..., N\}$, $\mathbf{x}'_i$ is the i-th column of $\mathbf{X}'$ and also the updated feature vector of node i.

We empirically find using different weight matrices for the self-node and neighbors can significantly improve the performance:

$$\mathbf{x}'_i = \sigma(\mathbf{Q}\mathbf{x}_i \hat{a}_{ii} + \sum_{j \in \mathcal{N}_i} \mathbf{W}\mathbf{x}_j \hat{a}_{ij}) \tag{3}$$

where $\mathbf{Q}$ is the weight matrix corresponding to the self-transformation. We will use this formulation as our baseline GCN in the experiments.

Fig. 2 demonstrates a graph convolutional layer for a simple 3-node graph and presents how each node is updated according to its neighbouring nodes. Within a single graph convolutional layer, only those nodes which are directly connected with a node could transmit information to it. There is no explicit mechanism for the GCN to learn global features that could be critical to 3D HPE. We will introduce our solution to this problem in the next section.

3.2 Learning Global Pose Features

Some global pose features, e.g., the action, viewpoint or scale of a person, can help reduce uncertainty in 3D HPE. For example, the action of a person, e.g.,

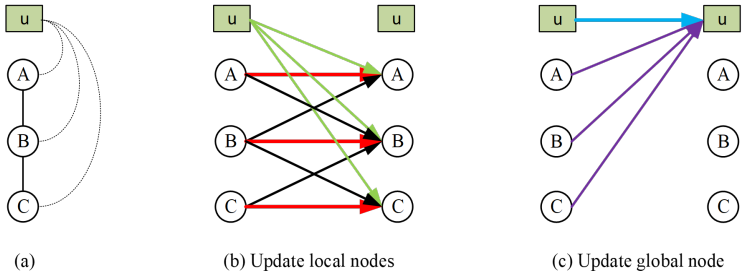

Fig. 3. Illustration of the feature update for local nodes and the global node. Arrays of the same color means applying the same weight matrix. (a) A simple graph composed of three nodes: A, B and C. **u** is the added global node, which is connected to all local nodes. (b) The feature update of local nodes. (c) The feature update of the global node.

walking or sitting, provides strong constraints on the relative locations of body joints, which eases pose estimation. This motivates us to learn global pose features in a GCN for 3D HPE.

Graph convolution with a global node. To achieve this goal, we add a global node to the graph and connect it to all local nodes, e.g., the body joint nodes. The global features are obtained by aggregating all body joint features to model the global pose attributes. The feature update of each body joint depends on not only their neighbors but also the global node. Specifically, a graph convolutional layer with a global node is defined as:

$$\mathbf{x}_i' = \sigma(\mathbf{Q}\mathbf{x}_i\hat{a}_{ii} + \sum_{j\in\mathcal{N}_i}\mathbf{W}\mathbf{x}_j\hat{a}_{ij} + \mathbf{T}\mathbf{x}_u) \tag{4}$$

$$\mathbf{x}_u' = \sigma(\frac{1}{N}\sum_{j=1}^{N}\mathbf{R}\mathbf{x}_j + \mathbf{S}\mathbf{x}_u) \tag{5}$$

where $i \in \{1, \cdots, N\}$ indexes a local node and u represents the global node, $\mathbf{Q}, \mathbf{W}, \mathbf{T}, \mathbf{R}, \mathbf{S}$ are different weight matrices. Since the global node and local nodes carry different types of features, we assign different transformation matrices to them.

Eq. (4) is the feature update rule for local nodes, which is the summation of three terms. The first term is the feature transformation of the i-th node itself, corresponding to the self-connection. The second term aggregates the transformed features of the neighboring nodes. The last term transforms the global features. Eq. (5) is the update function for global features, which sums up two terms. The first term aggregates features from all local nodes. The second term is the feature transformation of the global node itself, corresponding to the self-connection of the global node.

Fig. 3 demonstrates the feature updates of local nodes and the global node. The global node takes information from all local nodes and also contributes

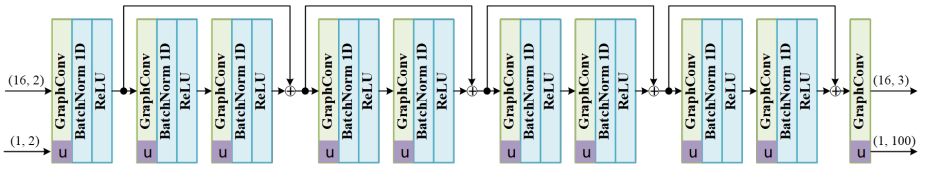

Fig. 4. This figure shows the GCN architecture we apply in our experiments. The building block is a residual block composed of two graph convolutional layers with 128 channels. This block is repeated four times. Each graph convolutional layer (except for the last one) is followed by a batch normalization layer and a ReLU activation layer.

to the feature update of local nodes. This gives local nodes access to global information during inference.

Heterogeneous multi-task learning. While the output of each local node is supervised by the 3D coordinate of the corresponding body joint, it remains unclear how to deal with the output of the global node during training. One solution is to simply ignore the update of global features in the last layer. In this case, gradients could still propagate from the loss of local nodes to the global features in previous layers to update the weights during training. This is because the update of local features at the current layer relies on the global features at the previous layer. During inference, we only need to check the output of the local nodes to get the 3D human pose prediction. But one potential limitation of this solution is that the lack of supervision for the global node may lead to inferior learning of global pose features.

To address this problem, we propose a heterogeneous multi-task learning framework. While local nodes still output 3D locations of body joints, the global node is responsible for an auxiliary task. Note the auxiliary task should be related to the main task, i.e., 3D HPE, and facilitate the learning of global pose features. In this paper, we consider two different kinds of auxiliary tasks: action classification and pose embedding.

Action classification means to classify the 3D pose to be predicted into an action label, e.g., running or jumping. The output of the global node is the probability distribution over action classes. Due to the lack of action annotations in the dataset, we need to generate some pseudo labels. Specifically, we use K-means to cluster the ground truth 3D poses in the training data to obtain their class labels and each cluster corresponds to an action class. Then, these action labels can be used to supervise the output of the global node during training, e.g., via a cross-entropy loss.

Pose embedding means to learn a low-dimensional representation of the pose. The output of the global node is a real-value embedding vector. We use a 2-layer decoder to reconstruct the 3D human pose from the embedding. The reconstruction error serves as the loss function of the global node. The decoder network is learned end-to-end with the embedding.

Table 1. Ablation study on the effectiveness of learning global pose features. **GN** is the abbreviation of global node. The supervision of the global node is through the **action classification** task (100 action classes). **Unsupervised GN** means the global node output is directly discarded in the training phase while **supervised GN** means the global node is supervised by action labels generated by K-means. All errors are measured in millimeters (mm).

Method	Channels	Params	MPJPE	P-MPJPE	Loss
Baseline GCN	205	0.69 M	41.87	33.53	**0.000079**
GCN (w/ **unsupervised GN**)	128	0.69 M	41.44	31.83	0.000124
GCN (w/ **supervised GN**)	128	0.69 M	**40.44**	**31.38**	0.000165
Baseline GCN	410	2.71 M	41.73	32.56	**0.000036**
GCN (w/ **unsupervised GN**)	256	2.69 M	41.27	31.16	0.000058
GCN (w/ **supervised GN**)	256	2.69 M	**38.72**	**30.75**	0.00009

3.3 Network Architecture

We adopt the network architecture shown in Fig. 4 for 3D HPE. The action classification is taken as the auxiliary task here. Following Martinez et al. [1] and Defferrard et al. [33], we stack multiple cascaded blocks, each of which is made up of two graph convolutional layers interleaved with batch normalization and ReLU. After that, we wrap every block as a residual block. Both the input and the output of the GCN are composed of two parts corresponding to local nodes and the global node, respectively. Specifically, the input is the 2D coordinates of the body joints and a zero-initialized vector. The output is the 3D body locations and the pose classification result. The overall loss is a summation of an $L2$-norm loss for 3D HPE and another loss for the auxiliary task, i.e., a cross-entropy loss for action classification and an $L2$-norm loss for pose embedding.

4 Experiments

4.1 Datasets and Evaluation Protocols

We conduct our experiments on the widely used dataset Human3.6M [34] and dataset MPI-INF-3DHP [35], and follow the previously used evaluation methods.

Human3.6M. This is the most popular indoor dataset for 3D HPE. It contains 3.6 million images filmed by 4 synchronized high-resolution progressive scan cameras at 50 Hz [34]. There are 11 subjects in total performing 15 daily activities such as walking, sitting, greeting and waiting. However, only 7 subjects are annotated with 3D poses. For fair comparison, we follow previous work [23, 36, 11], i.e. 5 subjects (S1, S5, S6, S7, S8) of the 7 annotated subjects are used for training while the rest 2 subjects (S9 and S11) are used for testing. We train and test our GCN models on all 15 actions.

Table 2. Ablation study on the number of action classes. **GN** is the abbreviation of global node. The supervision of the global node is through the **action classification** task. The column of **Classes** indicates different numbers of action classes. **Supervised GN** means the global node is supervised by action labels generated by K-means. All errors are measured in millimeters (mm).

Method	Classes	Channels	Params	MPJPE	P-MPJPE
GCN (w/ **supervised GN**)	50	128	0.69 M	40.95	31.67
GCN (w/ **supervised GN**)	100	128	0.69 M	40.44	**31.38**
GCN (w/ **supervised GN**)	200	128	0.71 M	**40.25**	31.60
GCN (w/ **supervised GN**)	50	256	2.66 M	40.03	30.98
GCN (w/ **supervised GN**)	100	256	2.69 M	**38.72**	**30.75**
GCN (w/ **supervised GN**)	200	256	2.74 M	39.82	30.91

Two protocols are widely used for evaluation. **Protocol-1** is the mean per-joint position error (MPJPE), which computes the averaged Euclidean distance error per joint between the prediction and the corresponding ground truth in millimeters. **Protocol-2** computes the same error after the alignment of the root joint of the prediction in accordance with the ground truth using a rigid transformation. The abbreviation of Protocol-2 is P-MPJPE.

MPI-INF-3DHP. This dataset is constructed by the Mocap system, containing both indoor and outdoor scenes with 3D pose annotations. We dismiss its training set, and only use the test set consisting of 2929 frames from six subjects conducting seven actions to evaluate the generalization capacity of our model. The results from this dataset are reported using the metrics 3D PCK and AUC [35].

4.2 Ablation Study

To avoid the influence of 2D human pose detector, we use 2D ground truth as the input for local nodes and initialize the global node with a zero vector. We adopt Adam [37] as the optimization method with an initial learning rate 0.001 and a decay rate 0.96 every 100K iterations. We initialize weights of GCNs using the method introduced in [38]. Following Zhao et al. [11], we set 128 as the default number of channels of each graph convolutional layer. We choose the optimal weight of the auxiliary loss via cross-validation: 0.001 for the cross-entropy loss used in action classification and 0.0001 for the $L2$-norm loss used in pose embedding. Eq. (3) is taken as our baseline GCN.

Learning global pose features. We first merely add a global node (**GN**) to our baseline GCN. When training and testing the GCN, there is no supervision for the global node. Then, we add the auxiliary task of action classification to supervise the learning of global features. Tab. 1 shows the results. To make sure that all models have the same number of parameters, we increase the number of

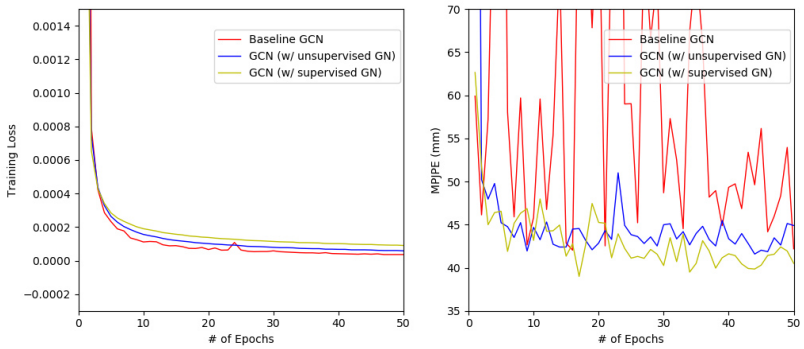

Fig. 5. The trend of the 3D HPE loss and the validation MPJPE during training. The number of parameters of each model is approximately 2.69M. **Unsupervised GN** means the global node output is directly discarded in the training phase while **supervised GN** means the global node is supervised by action labels generated by K-means.

channels for the baseline GCN. We can see that the baseline GCN has the lowest regression loss, but its error is higher than other two models. We infer this is mainly caused by overfitting. In this table, the GCN with a supervised global node performs the best given the same number of parameters. As we double the feature channels of hidden layers, the trend is more obvious. Thus, merely adding a global node could improve the performance of the baseline GCN, especially in P-MPJPE. With supervision, the GCN with a global node could perform better both in Protocol-1 and Protocol-2. The results shown in this table verify the effectiveness of both the global node and its supervision.

Furthermore, we plot the training loss and 3D HPE error descending curves, as shown in Fig. 5. Here, the training loss corresponds to the 3D HPE part of total training loss, excluding the loss from the global node. We can see from the figure that the training loss is becoming higher as we add a global node and then its supervision. The reason behind this is that adding a global node and its supervision would increase the importance of the global node, forcing the model to optimize global features. And we can see from the right-sided figure that the 3D HPE error is smaller and more stable when we add a global node and then the global node supervision. These results indicate that learning the global features and the auxiliary task improves the generalization ability of the GCN on 3D HPE.

The auxiliary task of action classification. We use action classification as the auxiliary task to supervise the global node. The output of the global node is a probability distribution on the action classes. We cluster all 3D poses in the training set into 50, 100 and 200 action classes, respectively. A larger number of action classes generally leads to a more difficult classification task. We visualize some clustering centers in Fig. 6. Obviously, these actions are very different

Table 3. Ablation study on the embedding dimension. **GN** is the abbreviation of global node. The supervision of the global node is through the **pose embedding** task. The column of **Embedding** indicates different embedding feature dimensions. **Supervised GN** means the global node is supervised by the reconstruction loss of the embedding. All errors are measured in millimeters (mm).

Method	Embedding	Channels	Params	MPJPE	P-MPJPE
GCN (w/ **supervised GN**)	10	128	0.67 M	**40.52**	**31.42**
GCN (w/ **supervised GN**)	20	128	0.67 M	40.73	31.62

Table 4. Quantitative comparisons on the Human 3.6M dataset under **Protocol-1**. The MPJPEs are reported in millimeters. The best results are highlighted in bold. **Legend:** (+) uses extra data from MPII dataset. (†) uses temporal information. (*) uses pose scales in both training and testing.

Protocol # 1	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Hossain et al. [7] ECCV'18 (†)	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
Pavllo et al. [36] CVPR'19 (†)	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. [5] ICCV'19 (†)	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
Pavlakos et al. [3] CVPR'17 (*)	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Martinez et al. [1] ICCV'17	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
Tekin et al. [39] ICCV'17	54.2	61.4	60.2	61.2	79.4	78.3	63.1	81.6	70.1	107.3	69.3	70.3	74.3	51.8	63.2	69.7
Yang et al. [20] CVPR'18 (+)	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
Pavlakos et al. [16] CVPR'18 (+)	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
Fang et al. [40] AAAI'18	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
Zhao et al. [11] CVPR'19	48.2	60.8	51.8	64.0	64.6	53.6	51.1	67.4	88.7	57.7	73.2	65.6	48.9	64.8	51.9	60.8
Sharma et al. [41] ICCV'19	48.6	54.5	54.2	55.7	62.2	72.0	50.5	54.3	70.0	78.3	58.1	55.4	61.4	45.2	49.7	58.0
Ci et al. [23] ICCV'19 (+)(*)	46.8	52.3	44.7	50.4	52.9	68.9	49.6	46.4	60.2	78.9	51.2	50.0	54.8	40.4	43.3	**52.7**
Ours	48.4	53.6	49.6	53.6	57.3	70.6	51.8	50.7	62.8	74.1	54.1	52.6	58.2	41.5	45.0	54.9

from each other: some of them are sitting while some of them are standing. We compare the performance of GCNs whose auxiliary task is to classify different numbers of action categories. The results are shown in Tab. 2. We find that when the number of feature channels is relatively small, the performance of these GCNs is robust to the number of action classes. But when the number of feature channels is doubled, categorizing poses into 100 classes helps the 3D HPE the most.

The auxiliary task of pose embedding. We also consider pose embedding as an auxiliary task. The output of the global node is an embedding vector whose dimension is smaller than that of a 3D pose vector (48 for 16 body joints). In our experiments, we compare the results obtained by setting the embedding dimension to 10 and 20, respectively. Tab. 3 shows that the dimension of the embedding only affects the performance slightly.

Comparing Tab. 2 and Tab. 3, using different auxiliary tasks affects the performance differently. Generally, the auxiliary task of action classification is more

Table 5. Quantitative comparisons on the Human 3.6M dataset under **Protocol-2**. The P-MPJPEs are reported in millimeters. The best results are highlighted in bold. **Legend:** (+) uses extra data from MPII dataset. (†) uses temporal information. (*) uses pose scales in both training and testing.

Protocol # 2	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Hossain et al. [7] ECCV'18 (†)	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
Pavllo et al. [36] CVPR'19 (†)	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. [5] ICCV'19 (†)	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
Sun et al. [18] ICCV'17	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
Martinez et al. [1] ICCV'17	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
Fang et al. [40] AAAI'18	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
Li et al. [9] CVPR'19	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. [23] ICCV'19 (+)(*)	36.9	41.6	38.0	41.0	41.9	51.1	38.2	37.6	49.1	62.1	43.1	39.9	43.5	32.2	37.0	**42.2**
Ours	38.4	41.1	40.6	42.8	43.5	51.6	39.5	37.6	49.7	58.1	43.2	39.2	45.2	32.8	38.1	42.8

Table 6. Quantitative comparisons on the Human 3.6M dataset under **Protocol-1**. All approaches take 2D ground truth as input. The MPJPEs are reported in millimeters. **Legend:** (+) uses extra data from MPII dataset. (*) uses pose scales in both training and testing.

Protocol # 1	Dire.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD.	Smoke	Wait	WalkD.	Walk	WalkT.	Avg.
Zhou et al. [10] ICCV'19 (+)	34.4	42.4	36.6	42.1	38.2	39.8	34.7	40.2	45.6	60.8	39.0	42.6	42.0	29.8	31.7	39.9
Ci et al. [23] ICCV'19 (+)(*)	36.3	38.8	29.7	37.8	34.6	42.5	39.8	32.5	36.2	39.5	34.4	38.4	38.2	31.3	34.2	36.3
Martinez et al. [1] ICCV'2017	37.7	44.4	40.3	42.1	48.2	54.9	44.4	42.1	54.6	58.0	45.1	46.4	47.6	36.4	40.4	45.5
Zhao et al. [11] CVPR'19	37.8	49.4	37.6	40.9	45.1	**41.4**	40.1	48.3	50.1	**42.2**	53.5	44.3	40.5	47.3	39.0	43.8
Ours	**36.2**	**40.8**	**33.9**	**36.4**	**38.3**	47.3	**39.9**	**34.5**	**41.3**	50.8	**38.1**	**40.1**	**40.0**	**30.3**	**33.0**	**38.7**

beneficial to the task of 3D HPE and the semantic meaning of the global node output is clear, but we need to generate pseudo action class labels by ourselves. As for the auxiliary task of pose embedding, it does not require extra generated labels. However, the output of the global node does not have explicit meanings and an extra decoder network which is composed of simple fully connected layers is needed. In addition, the global node output can also be used for other purposes. For example, it can be pose features for the task of human shape restoration or action recognition, for which pose information is significant.

4.3 Comparison with the State of the Art

Results on Human3.6M Following Pavllo et al. [36], we use 2D poses provided by a pre-trained 2D pose detector composed of cascaded pyramid network (CPN) [42] for benchmark evaluation. We use the GCN with a global node and an auxiliary task of 100-category action classification due to its overall best performance. We set the initial learning rate as 0.001, the decay factor 0.95 per 4 epochs and the batch size 256. We also apply dropout with a factor of 0.2 for

Table 7. Quantitative comparisons on the MPI-INF-3DHP dataset. The auxiliary task for the global node in our method is 100-class pose classification.

	Training Data	**GS** (PCK)	**noGS** (PCK)	**Outdoor** (PCK)	**ALL** (PCK)	**ALL** (AUC)
Martinez et al. [1]	H36m	49.8	42.5	31.2	42.5	17.0
Yang et al. [20]	H36m+MPII	-	-	-	69.0	32.0
Zhou et al. [17]	H36m+MPII	71.1	64.7	72.7	69.2	32.5
Pavlakos et al. [16]	H36m+MPII+LSP	76.5	63.1	77.5	71.9	35.3
Ci et al. [23]	H36m	74.8	70.8	77.3	74.0	36.7
Wang et al. [12]	H36m	-	-	-	71.9	35.8
Li et al. [9]	H36m+MPII	70.1	68.2	66.6	67.9	-
Zhou et al. [10]	H36m+MPII	75.6	71.3	**80.3**	75.3	38.0
Ours	H36m	**79.0**	**79.3**	79.8	**79.3**	**45.9**

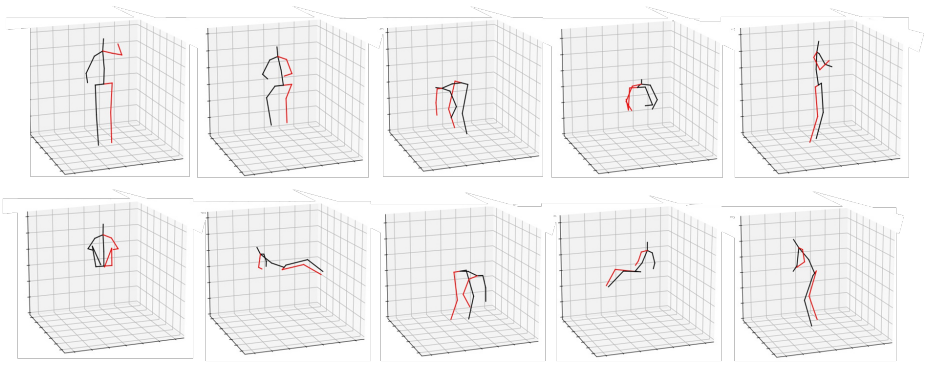

Fig. 6. These 3D poses are visualized K-means clustering centers when we categorize poses in the training set of Human3.6M into 50 action classes. Each pose category roughly represents a typical action, like waving, bending, lying and so on.

each graph convolutional layer. It takes about 4 hours to train our model for 50 epochs on a single GPU of Nvidia RTX 2080Ti. Tab. 4 and Tab. 5 compare our results and other state-of-the-art results under two protocols, respectively. In Protocol-1, the 3D pose error of our method is 54.9mm, which is lower than many recent state of the arts [40, 11, 41]. When trained on ground-truth 2D poses, our model outperforms other methods [1, 11] by a notable margin, as shown in Tab. 6. In Protocol-2, our method is comparable with previous state of the art [23] despite they use extra data from MPII dataset for training and exploit the information of pose scale in both training and testing. Note that we do not incorporate any additional modules, such as non-local [11, 5] and pose refinement [5], to further boost the performance of our method in these two protocols. In addition, the global node output in our model could be employed for

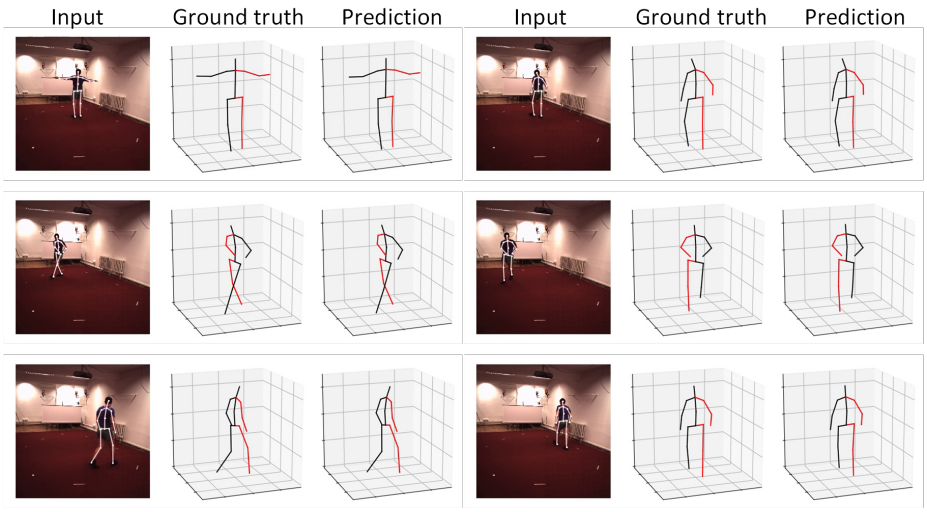

Fig. 7. Some qualitative results of our approach on Human3.6M.

follow-on tasks, like action recognition. Some qualitative results of our approach on Human3.6M dataset are presented in Fig. 7.

Results on MPI-INF-3DHP Following [9], we apply our model trained on the training set of Human3.6M to the test set of MPI-INF-3DHP. The 2D joints provided by the dataset are taken as input. Tab. 7 shows the results. As we can see from the table, our method outperforms other recent methods in "**GS**" and "**noGS**". Though [10] has slightly higher PCK in "**Outdoor**", overall our method achieves the best performance in contrast with previous state of the arts [23, 10, 16, 9] which attempt to address the generalization issue across different datasets. Notably, some of them even use more than one dataset to train their models. Since our model has not seen any pose contained in MPI-INF-3DHP, the results validate the generalization capacity of our model to new datasets.

5 Conclusion

In this paper, we introduce a novel 3D HPE approach by learning global pose features for 3D HPE. We also propose a heterogeneous multi-task learning framework to facilitate the learning of global features. With extensive ablation study and benchmark comparison, we make the following conclusions. (1) A global node is beneficial to GCNs for 3D HPE. (2) With supervision, a global node could learn global features better. (3) Both auxiliary pose classification and pose embedding are helpful to the supervision of a global node.

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. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3d human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 2640–2649
2. Moon, G., Chang, J.Y., Lee, K.M.: Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 10133–10142
3. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3d human pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 7025–7034
4. Qiu, H., Wang, C., Wang, J., Wang, N., Zeng, W.: Cross view fusion for 3d human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 4342–4351
5. Cai, Y., Ge, L., Liu, J., Cai, J., Cham, T.J., Yuan, J., Thalmann, N.M.: Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 2272–2281
6. Arnab, A., Doersch, C., Zisserman, A.: Exploiting temporal context for 3d human pose estimation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3395–3404
7. Rayat Imtiaz Hossain, M., Little, J.J.: Exploiting temporal information for 3d human pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 68–84
8. Dong, J., Jiang, W., Huang, Q., Bao, H., Zhou, X.: Fast and robust multi-person 3d pose estimation from multiple views. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 7792–7801
9. Li, C., Lee, G.H.: Generating multiple hypotheses for 3d human pose estimation with mixture density network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 9887–9895
10. Zhou, K., Han, X., Jiang, N., Jia, K., Lu, J.: Hemlets pose: Learning part-centric heatmap triplets for accurate 3d human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 2344–2353
11. Zhao, L., Peng, X., Tian, Y., Kapadia, M., Metaxas, D.N.: Semantic graph convolutional networks for 3d human pose regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 3425–3435
12. Wang, J., Huang, S., Wang, X., Tao, D.: 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. (2019) 7771–7780
13. Tekin, B., Katircioglu, I., Salzmann, M., Lepetit, V., Fua, P.: Structured prediction of 3d human pose with deep neural networks. arXiv preprint arXiv:1605.05180 (2016)
14. Agarwal, A., Triggs, B.: 3d human pose from silhouettes by relevance vector regression. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. Volume 2., IEEE (2004) II–II
15. Zhao, X., Ning, H., Liu, Y., Huang, T.: Discriminative estimation of 3d human pose using gaussian processes. In: 2008 19th International Conference on Pattern Recognition, IEEE (2008) 1–4
16. Pavlakos, G., Zhou, X., Daniilidis, K.: Ordinal depth supervision for 3d human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 7307–7316

17. Zhou, X., Huang, Q., Sun, X., Xue, X., Wei, Y.: Towards 3d human pose estimation in the wild: a weakly-supervised approach. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 398–407
18. Sun, X., Shang, J., Liang, S., Wei, Y.: Compositional human pose regression. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 2602–2611
19. Sun, X., Xiao, B., Wei, F., Liang, S., Wei, Y.: Integral human pose regression. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 529–545
20. Yang, W., Ouyang, W., Wang, X., Ren, J., Li, H., Wang, X.: 3d human pose estimation in the wild by adversarial learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 5255–5264
21. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
22. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434 (2018)
23. Ci, H., Wang, C., Ma, X., Wang, Y.: Optimizing network structure for 3d human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 2262–2271
24. Agarwal, A., Triggs, B.: Recovering 3d human pose from monocular images. IEEE transactions on pattern analysis and machine intelligence **28** (2005) 44–58
25. Rogez, G., Rihan, J., Ramalingam, S., Orrite, C., Torr, P.H.: Randomized trees for human pose detection. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2008) 1–8
26. Ionescu, C., Li, F., Sminchisescu, C.: Latent structured models for human pose estimation. In: 2011 International Conference on Computer Vision, IEEE (2011) 2220–2227
27. Chen, X., Lin, K.Y., Liu, W., Qian, C., Lin, L.: Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 10895–10904
28. Liu, K., Ding, R., Zou, Z., Wang, L., Tang, W.: 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. Zou, Z., Liu, K., Wang, L., Tang, W.: High-order graph convolutional networks for 3d human pose estimation. In: BMVC. (2020)
30. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org (2017) 1263–1272
31. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
32. Battaglia, P.W., Hamrick, J.B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al.: Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261 (2018)
33. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems. (2016) 3844–3852

34. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE transactions on pattern analysis and machine intelligence **36** (2013) 1325–1339
35. Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., Theobalt, C.: Monocular 3d human pose estimation in the wild using improved cnn supervision. In: 3D Vision (3DV), 2017 Fifth International Conference on, IEEE (2017)
36. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3d human pose estimation in video with temporal convolutions and semi-supervised training. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 7753–7762
37. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
38. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. (2010) 249–256
39. Tekin, B., Márquez-Neila, P., Salzmann, M., Fua, P.: Learning to fuse 2d and 3d image cues for monocular body pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 3941–3950
40. Fang, H.S., Xu, Y., Wang, W., Liu, X., Zhu, S.C.: Learning pose grammar to encode human body configuration for 3d pose estimation. In: Thirty-Second AAAI Conference on Artificial Intelligence. (2018)
41. Sharma, S., Varigonda, P.T., Bindal, P., Sharma, A., Jain, A.: Monocular 3d human pose estimation by generation and ordinal ranking. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 2325–2334
42. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 7103–7112