# Differentiable Hierarchical Graph Grouping for Multi-Person Pose Estimation

Sheng Jin[1,2][0000−0001−5736−7434], Wentao Liu[2†][0000−0001−6587−9878], Enze Xie[1], Wenhai Wang[3], Chen Qian[2], Wanli Ouyang[4], and Ping Luo[1]

[1] The University of Hong Kong [2] SenseTime Research
[3] Nanjing University [4] The University of Sydney
{jinsheng, liuwentao, qianchen}@sensetime.com
wanli.ouyang@sydney.edu.au, pluo@cs.hku.hk

**Abstract.** Multi-person pose estimation is challenging because it localizes body keypoints for multiple persons simultaneously. Previous methods can be divided into two streams, *i.e.* top-down and bottom-up methods. The top-down methods localize keypoints after human detection, while the bottom-up methods localize keypoints directly and then cluster/group them for different persons, which are generally more efficient than top-down methods. However, in existing bottom-up methods, the keypoint grouping is usually solved independently from keypoint detection, making them not end-to-end trainable and have sub-optimal performance. In this paper, we investigate a new perspective of human part grouping and reformulate it as a graph clustering task. Especially, we propose a novel differentiable Hierarchical Graph Grouping (HGG) method to learn the graph grouping in bottom-up multi-person pose estimation task. Moreover, HGG is easily embedded into main-stream bottom-up methods. It takes human keypoint candidates as graph nodes and clusters keypoints in a multi-layer graph neural network model. The modules of HGG can be trained end-to-end with the keypoint detection network and is able to supervise the grouping process in a hierarchical manner. To improve the discrimination of the clustering, we add a set of edge discriminators and macro-node discriminators. Extensive experiments on both COCO and OCHuman datasets demonstrate that the proposed method improves the performance of bottom-up pose estimation methods.

**Keywords:** Human Pose Estimation, Graph Neural Network, Grouping

## 1 Introduction

Multi-person pose estimation aims at localizing 2d keypoints of an unknown number of people in an image. It has attracted much research interest because of its significance in various real-world applications, such as human behavior understanding, human-computer interaction, and action recognition.

[†]Corresponding author.

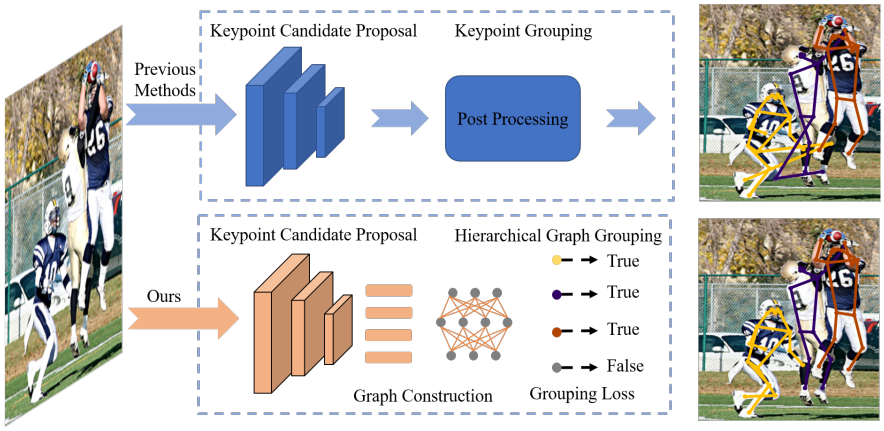



Fig. 1. Hierarchical Graph Grouping embeds grouping procedure with the keypoint candidate proposal network. All modules are differentiable and can be trained end-to-end. Keypoint candidates are grouped in a multi-layer graph neural network, which enables to directly supervise the final grouping results.

Current pose estimation methods perform keypoints detection in two routes. The *top-down* methods [6, 16, 26, 34, 38, 39, 46] first detect human bounding boxes and then estimate keypoints for each person. It performs a single person pose estimation to all human candidates, so it is often time-consuming. Contrarily, *bottom-up* pose estimation approaches [3, 22, 29, 33] follow the keypoints detection-and-grouping pipeline: detecting keypoints at the first stage and grouping them into individuals at the second stage. These methods are more efficient and have gained increasing attention in the industry. Previous works generally treat the grouping stage as post-processing by using integer linear programming [18, 19, 23, 35], heuristic greedy parsing [3, 33], or clustering [29, 31]. But they are not able to be trained end-to-end, which is in conflict with deep learning's philosophy of learning everything together. Previous bottom-up methods generally learn some substitute indicators which may reflect the grouping accuracy, resulting in sub-optimal solutions. For example, associate embedding (AE) [29] produces the permutation-invariant associative embedding (a vector representation) for each keypoint, and learns by pushing apart the embedding of different people and pulling closer that of the same instance. Although it uses the associative embedding which encodes pairwise relationship to group keypoints, the grouping procedure itself is still offline, and no direct supervision is applied to the grouping results. There is a mismatch between the pairwise loss and the accuracy of the greedy parsing used at inference time. Even though the pairwise loss is low, the parsing results can still be possibly wrong, and vice versa.

A better choice is to directly supervise the grouping process. However, one major challenge is that the previous keypoint grouping procedure is often not dif-

ferentiable, and thus is hard to be integrated with keypoint detection. Moreover, how to deal with the flexible number of keypoints is still an open problem.

In this paper, we present a simple and elegant solution for bottom-up multi-person pose estimation. In the proposed method, the whole network, composed of a keypoint detection network and a grouping network, is *fully end-to-end trainable*, and able to flexibly deal with the grouping problem of a variable number of human instances. To achieve this, we first reformulate the grouping problem as the graph clustering problem. A graph corresponds to an image, where the nodes denote the keypoint proposals, and edges denote whether the two keypoints belong to the same person. The graph structure is adaptive to different input images instead of constructing a static graph, so it is able to dynamically group various numbers of keypoints into various numbers of human instances. Especially, we propose the Online Hierarchical Graph Clustering (OHGC) algorithm, which makes the process of grouping keypoints learnable and can be easily embedded into main-stream bottom-up methods. The HGG method initializes the graph from the keypoint proposal network and groups pairs of most relative nodes in each iteration through the OHGC algorithm.

In OHGC, keypoints are clustered step-by-step. Each keypoint proposal starts in its own graph node, and the cluster pairs are merged. This forms a pose hierarchy, from small fractions to the whole body. This enables the model to pay more attention to global consistency and learn effective features for predicting the pairwise relation. The group operations are fully differentiable, so OHGC can make the whole network (including keypoint detection and grouping) end-to-end trainable. By directly supervising the grouping results, the grouping loss is back-propagated to the previous keypoint detection network, which will further improve the feature representation ability of the keypoint detection network.

Moreover, we propose the edge discriminator to strengthen the local relationship of keypoints, and the macro-node discriminator to enforce global consistency. It can further increase the discrimination of body-keypoint relational features, leading to better grouping accuracy.

The main contributions of this work are thus three-fold.

- We reformulate the task of multi-person pose estimation as a graph clustering problem and present the first fully end-to-end trainable framework with grouping supervision for bottom-up multi-person pose estimation.
- We propose edge discriminators and macro-node discriminators to learn both local and global pairwise relation features and boost the grouping accuracy.
- The experimental results show that the proposed method outperforms the baseline by a large margin and achieves comparable performance with the state-of-the-art bottom-up pose estimation methods on COCO dataset. Moreover, the proposed method achieves the state of the art performance on the OCHuman datasets (41.8/36.0 mAP for val and test respectively).

## 2 Related Work

### 2.1 Multi-person Pose Estimation in Images

**Top-down** methods [6, 12, 16, 17, 26, 28, 34, 38, 46] decompose the multi-person pose estimation task into two sub-tasks:(1) Human detection and (2) Pose Estimation in the region of a single human. First, the person detector predicts a bounding box for every human instance in the image. Second, the box is cropped and resized from the image. Third, single-person pose estimation is applied to predict the keypoints for the cropped person. In addition, some work such as Mask R-CNN [16] crop the feature instead of raw images to boost efficiency. In summary, top-down methods are dominant in state-of-the-art methods but they often have higher computational complexity overhead, especially when the number of human instances increases. This is because they need to repeatedly run the single-person pose estimation for every instance. Furthermore, because the pose estimation is dependent on the detection, it is difficult for these methods to recover the pose of an instance if it is missing in the detection results.

**Bottom-up** approaches [3, 18, 19, 21–23, 29, 31, 33, 35] first detect all keypoint candidates in an image, then assemble/group them into full-body keypoints of each instance. Such bottom-up methods are usually efficient, and are capable of achieving real-time performance. To aid the follow-up keypoint association, most bottom-up methods learn descriptors to encode keypoint pairwise relations and to distinguish different instances. PAF [3] learns part-affinity-fields, encoding both the location and orientation of keypoint pairs; GPN [31] learns 2D offset fields, linking keypoints to the corresponding human centers; PersonLab [33] introduces long-range, mid-range and short-range offsets between pairwise keypoints; AE [29] learns the associative embedding for each keypoint and similar embedding indicates higher possibility of belonging to the same person. The grouping process is generally formulated as a post-processing optimization problem and solved by graph partitioning [18, 19, 21, 35], heuristic greedy decoding algorithm [3, 33] or spectral clustering [31]. In summary, bottom-up methods can benefit from sharing convolutional computation, as a result, being faster than top-down methods. Nevertheless, the post-processing of grouping is heuristic and involves many hyper-parameters. Since the pose estimation and post-processing are not jointly learned, they cannot collaborate and adapt to each other. Instead of regarding the grouping as a pure post-processing procedure, we propose to train grouping with pose estimation jointly in an end-to-end fashion, enabling the error signals for grouping to be back-propagated.

**Single-stage pose estimation.** With recent advantages of single-shot object detection and instance segmentation [41, 47, 52], some single-stage pose estimation methods are proposed. CenterNet [52] firstly transfer pose estimation as human center detection and keypoint regression. However, it still needs keypoint detection and projection to improve performance. SPM [32] proposes a structured pose representation to divide the keypoints hierarchically. In this way, it can ease the difficulty of long-range regression. Similarly, DirectPose [40], based on FCOS [41], directly do human center classification and keypoint regression

without relying on bounding box. KPAlign is proposed to overcome the feature misalignment between convolutional features and keypoint predictions. However, keypoint regression is not very precise in the above methods, especially under the restriction of High IoU. In comparison, our method retains higher precision, especially under more strict metrics ($AP_{75}$).

### 2.2 Graph Representation for Pose Estimation

The graph representation for human pose estimation is not new. For single-person pose estimation, many work [4, 5, 8, 13, 14, 24, 42, 49] have been based on various graphical models such as pictorial structure, Mixtures-of-parts, Markov Random Fields (MRF) or Conditional Random Fields (CRF). In these works, the graph nodes represent keypoints and the edges encode the pairwise relationships between keypoints. Since all the keypoints belong to the same human instance, no grouping process is required. Moreover, the number of keypoints of a single person is always fixed, therefore the graph structure, in terms of the number of nodes and the connectivity of edges, is fixed.

Multi-person pose estimation is much more challenging. [18, 21, 44], the pose estimation problem is cast as a graph partitioning based integer linear programming (ILP) problem. However, the optimization process is offline and very time-consuming. Song *et al.* [37] proposed a method for end-to-end minimum cost multicut problem. Unlike their works which focus on the CRF optimization, we solve the keypoint grouping task by direct graph clustering.

### 2.3 Graph Neural Networks

This paper reformulates the multi-person pose estimation task using the graph representation and applies graph neural networks to this problem. Graph Neural Networks (GNN) is initially introduced in [15, 36] and has become a popular tool for efficient message passing and modeling global relations [7]. Most of GNN models can be categorized into two types: spectral approaches [2, 25] and non-spectral approaches [11, 45]. This work is related to [45], which efficiently models the edge features. To solve the task of multi-person pose estimation, based on [45] we develop a hierarchical clustering method, which takes the body structure constraints into consideration and models the whole grouping process.

More recently, GNN models have been applied to model the human body structure. Yan *et al.* [48] proposes the spatial-temporal graph convolutional networks for skeleton-based action recognition. Zhang *et al.* [50] proposes to use PGNN to learn the structured representation of keypoints for single-person pose estimation. However, previous works only deal with the single person case, where the structure of the graph is fixed. The multi-person case is more challenging, since the number of keypoints and the number of people vary in different images and even in different grouping stages. We have to develop a dynamic graph interaction model to effectively handle such problems.

## 3 Method

**Overview** An overview of our proposed hierarchical graph grouping (HGG) framework is illustrated in Fig 2. Our HGG framework consists of two stages, *i.e.* the keypoint candidate proposal stage and the keypoint grouping stage.

In the keypoint candidate proposal stage, all keypoint candidates are detected and corresponding feature maps are extracted. Following AE [29], we use a 4-stacked hourglass [30] as the backbone of the keypoint candidate proposal network. The keypoint proposal network then provides keypoint candidates and raw relational feature embedding for the keypoints grouping module.

In the keypoint grouping stage, we build a graph neural network using the candidates and relational features extracted from the former stage. An online hierarchical graph clustering (OHGC) algorithm is devised to cluster keypoints iteratively. In each iteration, OHGC updates the pairwise relation features and clusters nodes into a *macro-node* by maximizing the weighted edge score. The graph is updated and pruned with respect to the macro-nodes. Contrary to integer linear programming or bipartite matching, the proposed method is fully differentiable and is able to be trained end-to-end with keypoint detection.

We proposed two kinds of the discriminator to strengthen the grouping procedure, the edge discriminator and the macro-node discriminator. In each iteration, the edge discriminator is introduced to classify whether the pair of nodes belong to the same person. The pairwise relation features and the edge scores are updated accordingly. After each iteration of grouping, a macro-node discriminator is applied to each cluster to discriminate between a correctly-clustered macro-node (in which all nodes belong to the same person) and a wrongly-clustered one. In this way, the whole online grouping procedure is fully supervised.

### 3.1 Hierarchical Graph Grouping

Previous work [18, 19, 21, 35] cast the problem of multi-person pose grouping as graph partitioning, and solve it by optimizing an integer linear programming (ILP) problem. However, the optimization process is performed offline and the grouping procedure is not able to be supervised with the keypoint candidate proposal network. In this paper, we rethink this problem from the perspective of graph clustering and solve it with supervised learning. We follow the online agglomerative graph clustering setting. Each keypoint candidate starts with being its own cluster and closest pairs of clusters are merged iteratively. As a result, the keypoint candidates are grouped into several clusters, where each cluster contains all the keypoints of a single person. We are able to directly supervise the final grouping results. In the following sections, we will give a detailed description of the graph construction and hierarchical graph grouping.

**Graph Construction** We construct a graph on top of the keypoint candidate proposal network. In the graph $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$, the "vertices" $\{\mathbf{V}\} = \{v_i\}_{i=1:N}$ represent keypoint candidates and the "edges" $\{\mathbf{E}\} = \{e_{i_1,i_2}\}_{i_1=1:N,i_2=1:N}$ represent the pairwise relationship between the two candidates (the possibility of

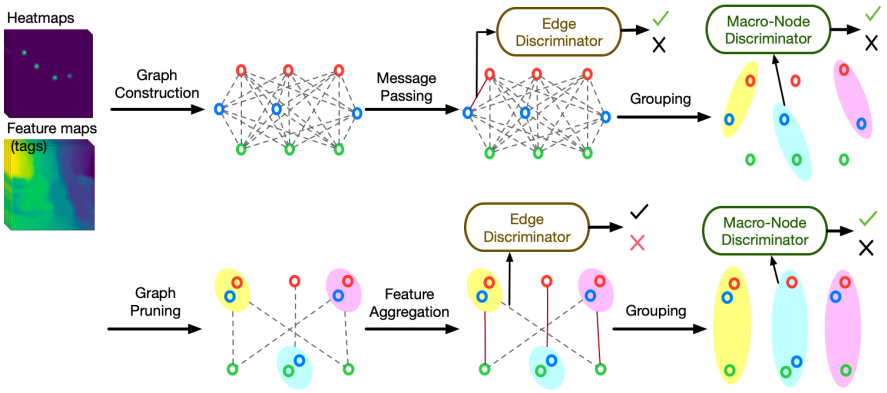



Fig. 2. The keypoint grouping stage of HGG framework. We construct a graph on top of the keypoint candidate proposal network, perform message passing with GNNs, and group the candidates iteratively. Edge discriminators and macro-node discriminators are applied to improve the grouping performance.

belonging to the same person or not). Note that the graph is constructed dynamically, as the graph may have different number of nodes and edges for different images. We choose the fully-connected graph that densely connects every pair of the keypoints with different keypoint types. The keypoints with the same type (both "head"s) are disconnected. Compared to other sparse graph configurations (such as the tree-structure), the fully-connected graph is able to avoid over-segment of a person during occlusion, *i.e.* dividing a single pose into several clusters. For example, when a person's torso is occluded or missing, the link between the head and the foot will be helpful to connect the upper and the lower parts. Moreover, since the number of keypoints in an image is only about 30 on average, the computational cost of constructing such a dense graph is almost negligible. Each vertex $v_i \in \{\mathbf{V}\}$ in the graph is initialized with the concatenation of the following features: (1) the relational embedding features of the keypoint, (2) the one hot feature that encodes the keypoint type, (3) the (x, y) coordinates of the keypoint normalized to $[0, 1]$. Both visual features and spatial features are preserved.

**Online Hierarchical Graph Clustering Algorithm** OHGC algorithm is given in Algorithm 1. Given the initial graph, an *interaction GNN* (Graph Neural Network) is trained to extract the relational features via message passing between vertices. As shown in Fig. 3, our GNN utilizes a stack of EdgeConv [45] layers for effective feature learning. In each EdgeConv layer, the edge feature is mapped from the concatenation of features of nodes (linked by the edge) using a fully-connected layer, and the node features are updated by aggregating the features of the associated edges. A three-layer MLP (Multi-layer Perceptron) with

Dropout is adopted to further extract high-level node features. As the output, we get representative features of each of the vertex which is used for grouping.

Previous graph clustering algorithms mainly focus on the keypoint-level pairwise relationship, without considering the higher-order term, *i.e.* the relation between two clusters of body parts. We instead propose to model the whole grouping process and design a hierarchical graph clustering algorithm. OHGC repeatedly performs graph feature aggregation, edge proximity update, node clustering and graph pruning, until all the edges are cut.

In each iteration, *feature aggregation* is applied to each of the macro-node (the set of previously grouped nodes) by averaging all features in the set. The proximity score between macro-nodes is measured by the edge discriminator (see Sec.3.2). After updating the edge weights, we use *graclus clustering* [9] to match each vertex with its neighbors by (approximately) maximizing the edge weights. This finds the most confident pairs and carries out the clustering action. As a result, a group of "low-level" nodes is clustered into a "higher-level" macro-node. The number of clusters is reduced by half. For COCO dataset, the number of keypoint types is $J = 17$, so the grouping will stop in no more than $\lceil \log_2 17 \rceil = 5$ iterations. After that, a macro-node discriminator (see Sec.3.2) is applied to each cluster to discriminate between a correctly-clustered macro-node (in which all nodes belong to the same person) and a wrongly-clustered one. The grouping procedure should satisfy the following two constraints. 1) A keypoint cannot be assigned to more than one person, *i.e.* two people share a single "head" keypoint. 2) A person cannot have more than one keypoints of the same type, *i.e.* a person containing two "head" keypoints. To avoid infeasible clustering, we perform *graph pruning* to remove infeasible edges after each grouping iteration. If two (macro-)nodes contain the same type of nodes, the edge in between is pruned. This grouping procedure repeats until all edges are pruned.

This grouping procedure naturally forms a hierarchy, from isolated keypoints to a whole body. The model learns to first group easy-to-group parts, then perform cluster in the macro-node level. As the grouping continues, the graph gradually gets coarsened. Finally, the nodes will be clustered into $K$ groups, indicating $K$ human instances. The model learns to group from easy to hard, in a curriculum fashion [1]. Unlike the previous curriculum learning paradigm which requires to manually set curriculum phases, our curriculum tasks are automatically generated during training and well adjusted to the model's current capability.

### 3.2 Grouping Discriminators

In OHGC, two types of discriminators are introduced to further improve the grouping performance. In each iteration, we utilize the edge discriminator to update proximity scores and the macro-node discriminator to suppress the incorrectly grouped macro-nodes. We use the same discriminators in each clustering loop iteration. The network architectures are demonstrated in Fig. 3. Binary cross-entropy (BCE) loss is used to train.

**Algorithm 1** Online Hierarchical Graph Clustering

**Input:** An RGB image;
**Output:** Body pose clusters;
Keypoint candidate proposals;
Graph construction: $\mathbf{G} = \{\mathbf{V}, \mathbf{E}\}$;
Relational feature learning with interaction GNN.
**repeat**
  Feature aggregation via avg-pooling;
  Update the proximity between (macro-)nodes;
  Apply graclus clustering;
  Graph pruning;
**until** No edges are remained.

**Edge Discriminator.** Edges preserve local but discriminate keypoint-to-keypoint relationship. In order to improve the discrimination ability of the pairwise relation feature, we introduce a *shared* edge discriminator at each iteration. The edge discriminator is a two-class discriminator that is used to directly classifying the states of the edges: whether the edge is connected (label 1) or not (label 0). Connected edge means the two keypoints belong to the same person. As shown in Fig. 3, the edge discriminator is implemented as a three-layer MLP (Multi-layer Perceptron) with Dropout. The input is the concatenated features of two linked (macro-)nodes ($2 \times 64 = 128-$D), and the output is the 1-D edge score. Experiments show that the edge discriminator helps to increase the discrimination of body-keypoint relational features, leading to better grouping accuracy.

**Macro-node Discriminator.** We propose the macro-node discriminator to directly supervise the grouping procedure. After each grouping iteration, the nodes are clustered into macro-nodes. We apply a *shared* macro-node discriminator to each macro-node to classify whether all keypoint candidates in the group belong to the same person (label 1) or not (label 0). Both the final human-level grouping results and the intermediate part-level grouping results are supervised. This provides denser supervision signals, facilitating the model training. The discriminator takes the aggregated macro-node features (64-D) as input and forwards it into a three-layer MLP to discriminate positive vs negative macro-nodes.

### 3.3 Implementation Details

**Keypoint Proposal Network.** The keypoint proposal network generates both 2D Gaussian confidence heatmaps as well as the pairwise relational feature maps. 2D Gaussian confidence heatmaps [3, 29, 31] are used to encode the keypoint locations and the ground truth confidence map for an image is calculated as the maximum of every person. We follow [3, 29] to apply keypoint NMS and parse the heatmaps to generate keypoint candidates. The pairwise relational feature maps are learned with push/pull losses, by pushing features of different people apart and pulling together features extracted from the same person.

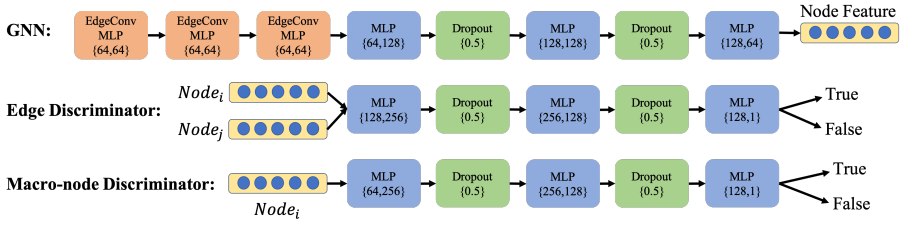



Fig. 3. The network architecture of GNN, the edge discriminator and the macro-edge discriminator. The number of the input/output channels of MLP are given.

**Training and Inference.** We implement OHGC based on AE [29][1]. The input size is set as $512 \times 512$ and the output size is $128 \times 128$. The keypoint proposal network is first pre-trained and the keypoint proposal network, GNN and the edge/macro-node discriminators are jointly trained in an end-to-end manner. The losses include keypoint detection loss, pairwise pull/push losses, binary cross-entropy (BCE) loss for discriminators. The weights to balance these losses are set as $1 : 1e^{-3} : 1e^{-5}$. We use Adam with an initial learning rate $2e^{-4}$ to train the model. During inference, flip testing and multi-scale testing is adopted. Unlike previous methods [3, 29], we do not use single-person refinement.

## 4 Experiments

### 4.1 Datasets and Evaluation

To verify the effectiveness of the proposed HGG, we compare it with state-of-the-art methods on two challenging datasets, i.e. MS-COCO [27], and OCHuman [51]. We follow [20] to use Average Percision (AP) to evaluate the methods.

**MS-COCO Dataset** [27] contains over 200,000 images and 250,000 human instances and 1.7 million labeled keypoints in total, among which 150,000 instances are for training and 80,000 instances are for testing. Our models are trained on the train set only. The ablation studies are reported on the val set and the comparisons with other state-of-the-arts are reported on the test-dev.

**OCHuman Dataset** [51] is a recently proposed benchmark to examine the limitations of human pose detection in highly challenging scenarios, which does not contain training samples and is intended to be used for evaluating existing models. It consists of 4731 images for validation and 8110 images for testing. The dataset contains only challenging cases of occlusion and the average IoU of the bounding boxes is 67%. Following [51], we train models on the training set of MS-COCO, and report the AP of them.

[1] https://github.com/princeton-vl/pose-ae-train

Table 1. (a) Comparisons with both top-down and bottom-up methods on COCO2017 test-dev dataset. * means using single-person pose refinement. × means using extra segmentation annotation. + means using multi-scale test. Not that our results are obtained without single-person pose refinement.(b) Comparisons with both top-down and bottom-up methods on OCHuman dataset. Our results are obtained without single-person pose refinement.

| Method | AP | $AP^{50}$ | $AP^{75}$ | $AP^{M}$ | $AP^{L}$ | AR |
|---|---|---|---|---|---|---|
| *Top-down methods* | | | | | | |
| Mask-RCNN [16] | 63.1 | 87.3 | 68.7 | 57.8 | 71.4 | – |
| G-RMI [34] | 64.9 | 85.5 | 71.3 | 62.3 | 70.0 | 69.7 |
| IPR [39] | 67.8 | 88.2 | 74.8 | 63.9 | 74.0 | – |
| CPN [6] | 72.1 | 91.4 | 80.0 | 68.7 | 77.2 | 78.5 |
| RMPE [12] | 72.3 | 89.2 | 79.1 | 68.0 | 78.6 | – |
| CFN [17] | 72.6 | 86.1 | 69.7 | 78.3 | 64.1 | – |
| SBL [46] | 73.7 | 91.9 | 81.1 | 70.3 | 80.0 | 79.0 |
| HRNet-W48 [38] | 75.5 | 92.5 | 83.3 | 71.9 | 81.5 | 80.5 |
| *Bottom-up methods* | | | | | | |
| OpenPose* [3] | 61.8 | 84.9 | 67.5 | 57.1 | 68.2 | 66.5 |
| AE*+ [29] | 65.5 | 86.8 | 72.3 | 60.6 | 72.6 | 70.2 |
| PersonLab+× [33] | 68.7 | 89.0 | 75.4 | 64.1 | 75.5 | 75.4 |
| Directpose+ [40] | 64.8 | 87.8 | 71.1 | 60.4 | 71.5 | – |
| SPM*+ [32] | 66.9 | 88.5 | 72.9 | 62.6 | 73.1 | – |
| Ours+ | 67.6 | 85.1 | 73.7 | 62.7 | 74.6 | 71.3 |

(a)

| OCHuman | Backbone | Val | Test |
|---|---|---|---|
| *Top-down methods* | | | |
| RMPE [12] | Hourglass | 38.8 | 30.7 |
| SBL [46] | ResNet50 | 37.8 | 30.4 |
| SBL [46] | ResNet152 | 41.0 | 33.3 |
| *Bottom-up methods* | | | |
| AE [29] | Hourglass | 32.1 | 29.5 |
| AE+ [29] | Hourglass | 40.0 | 32.8 |
| Ours | Hourglass | 35.6 | 34.8 |
| Ours+ | Hourglass | **41.8** | **36.0** |

(b)

### 4.2 Ablation Study

We validate the effectiveness of key modules in HGG by conducting the following ablation studies. For fair comparisons, all models use Hourglass as the backbone network and are trained with the same data augmentation and training schedule.

**Effectiveness of End-to-End Learning.** We compare the performance of the baseline Associate Embedding (AE) model and that with the grouping loss. The grouping loss is provided by the final level macro-node discriminator. As shown in Table 2 #1 and #3, end-to-end learning can increase the AP and the AR of the baseline by 0.6% and 1.3% respectively. #6 uses all these grouping losses to train the models, but uses original post-processing greedy grouping during inference. The improvement of #6 over #1 indicates that the grouping loss and end-to-end learning can improve the capability of Keypoint Proposal Network. Note that under this setting, the grouping module can be removed during inference without adding any additional computation overhead.

**Effectiveness of the Edge Discriminator.** The edge discriminator can enhance the keypoint relational features, thereby improving the grouping accuracy. To verify this, we compare the performance of models with and without the edge discriminator. As shown in Table 2 #1 and #4, we find that supervising the linkage of the edge will significantly improve the grouping performance by 2.0 mAP, demonstrating the effectiveness of the edge discriminator.

**Effectiveness of the Macro-Node Discriminator.** We evaluate two kinds of macro-node supervision, intermediate macro-node supervision and final macro-node supervision. As shown in #4 and #5, the final macro-node supervision improves the grouping performance by 0.5 mAP. By performing intermediate supervision to the macro-node, the result is further improved by 0.3 mAP, shown

Table 2. Ablation study of HGG's components on the COCO validation dataset. "FinalM" means the final level macro-node discriminator. "Edge" means edge discriminator. " IntermM" means intermediate macro-node discriminator. "MS" means multi-scale testing.

| # | Method | Clustering | FinalM | Edge | IntermM. | MS | AP | $AP^{50}$ | $AP^{75}$ | AR | $AR^{50}$ | $AR^{75}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | AE [29] | | | | | | 57.6 | 79.7 | 62.6 | 62.1 | 81.4 | 66.1 |
| 2 | AE [29] | | | | | ✓ | 65.6 | 85.1 | 71.9 | 69.1 | 86.7 | 74.2 |
| 3 | Ours | ✓ | ✓ | | | | 58.2 | 80.8 | 63.9 | 63.4 | 83.5 | 68.0 |
| 4 | Ours | ✓ | | ✓ | | | 59.6 | 81.3 | 65.1 | 64.2 | 83.0 | 69.0 |
| 5 | Ours | ✓ | ✓ | ✓ | | | 60.1 | 81.6 | 66.0 | 64.5 | 83.4 | 69.6 |
| 6 | Ours | | ✓ | ✓ | ✓ | | 59.6 | 81.9 | 65.5 | 63.9 | 83.3 | 68.4 |
| 7 | Ours-FC | ✓ | ✓ | ✓ | ✓ | | 58.3 | 80.7 | 63.3 | 62.5 | 82.1 | 66.9 |
| 8 | Ours-GAT | ✓ | ✓ | ✓ | ✓ | | 59.3 | 81.1 | 65.5 | 63.9 | 82.8 | 69.0 |
| 9 | Ours | ✓ | ✓ | ✓ | ✓ | | 60.4 | 83.0 | 66.2 | 64.8 | 84.0 | 69.8 |
| 10 | Ours | ✓ | ✓ | ✓ | ✓ | ✓ | 68.3 | 86.7 | 75.8 | 72.0 | 88.3 | 78.0 |

in #5 and #9. In total, the full supervision boosts the performance by 0.8 mAP, showing the importance of supervising the whole grouping process.

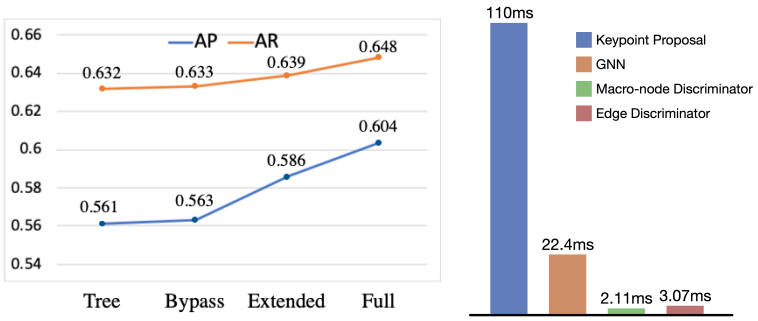
**Effectiveness of GNN.** To evaluate the interaction GNN, we add two baselines for comparison. Ours-GAT uses GAT [43], a popular graph neural network, for replacing EdgeConv. Ours-FC uses the multi-layer perception (dubbed FC for fully connected layers). For fair comparisons, these models have approximately the same parameter counts. As shown in #7, #8 and #9, both graph-based models perform better than Ours-FC baseline, because of more effective interactive message passing. Moreover, EdgeConv (60.4 AP) performs the best.

**Comparisons of Different Graph Configurations.** As shown in Fig 4a, four types of commonly used graph configurations [10] (*i.e.* Tree, Bypass, Extended and Full) are compared. From Tree (the standard tree-structured model) to Full (the fully-connected graph), the graph gets denser. Bypass and Extended model adds some skip connections to the standard tree-structured model. As the complexity of the graph (or the number of connections) increases (Tree-Bypass-Extended-Full), the grouping accuracy increases from 56.1% to 60.4% mAP. In addition, the runtime of different graph configurations is almost the same. Therefore, we choose the fully-connected graph in our implementation.

### 4.3 Qualitative Analysis

In Figure 5, we visualize the grouping procedure of OHGC algorithm. We use different colors to denote different clusters and dashed lines to highlight the macro-node merging process. OHGC starts with a set of keypoint candidates, each of which belongs to its own cluster. The grouping is performed iteratively. In each iteration, the most easy-to-group keypoints are merged. We show that the grouping procedure forms a pose hierarchy, from part to whole. Our method benefits from global supervision, which helps improve the grouping performance.

For failure cases, however, the current model is not able to recover false negatives or localization errors. Tiny people in images can lead to false negatives. Severe occlusion and non-typical poses may lead to localization errors. More test-time augmentation such as multi-scale testing, may mitigate these issues.




(a) (b)

Fig. 4. (a) Comparisons of different graph configurations on the COCO val set. Fully-connected graph (Full) performs the best among them. (b) Runtime analysis measured on one GTX-1060 GPU. The grouping module is very efficient compared to the keypoint proposal module.

### 4.4 Comparisons with the State-of-the-art Methods

We compare our framework with the state-of-the-art methods on two large-scale multi-person pose estimation benchmarks.

**Results on MSCOCO dataset** Table 1a shows experimental results on MSCOCO `test-dev` set. We see that the proposed HGG model achieves overall 67.6 AP. which is slightly lower than the state-of-the-art method PersonLab [33]. However, PersonLab uses extra annotations for instance segmentation. Moreover, we also compare our method with recent single-shot methods (SPM [32] and DirectPose [40]). Surprisingly, although ours are lower than them in $AP^{50}$, in $AP^{75}$ ours are superior to them. This further indicates that our methods have advantages in scenarios that require high-precision pose estimation.

**Results on OCHuman Dataset** To verify the robustness of HGG and other methods, we evaluate the proposed HGG model on the more challenging OCHuman dataset. We can see that our method achieves 41.8% and 36.0% mAP on val and test set, establishing a new state-of-the-art. Especially, HGG even outperforms top-down method SBL with 2.7 AP in test set, which further indicates our method is robust on more challenging scenarios.

### 4.5 Runtime Analysis

We analyze the time cost of the modules in HGG. Specifically, we evaluate our method on val set of MS-COCO and calculate the average time cost per image as shown in Fig. 4b. The results are tested using PyTorch with a batchsize of 1 on one GTX-1060 GPU in a single thread. We find that the time cost of the grouping module is only a small proportion of the total time cost.

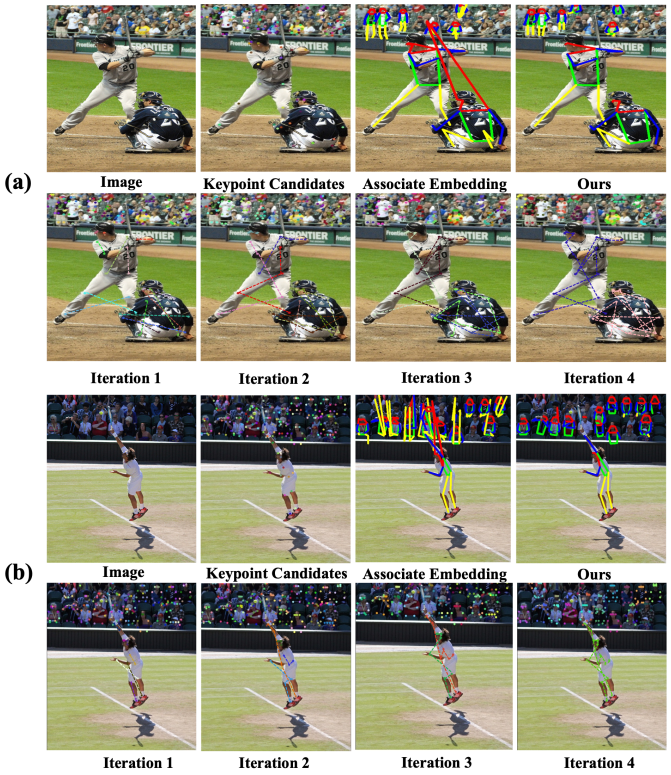



Fig. 5. The grouping process visualization. We show the grouped keypoint clusters in each iteration. Different colors are used to indicate different clusters.

## 5 Conclusion and Future Work

In this paper, we have reformulated the human pose estimation problem using the graph model and presented a full end-to-end learning framework named HGG. We have shown how we can combine the representative feature learning ability of CNN and the efficient long-range message passing as well as the relational feature learning capability of GNN. The macro-node discriminator and the edge discriminator are introduced to supervise the whole grouping process. We envision that the proposed framework can also be applied to other related problems such as multi-object tracking and instance segmentation. We expect to see more research in this direction in the near future.

**Acknowledgement.** This work is partially supported by the SenseTime Donation for Research, HKU Seed Fund for Basic Research, Startup Fund, General Research Fund No.27208720, the Australian Research Council Grant DP200103223 and Australian Medical Research Future Fund MRFAI000085.

## References


1. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: International Conference on Machine Learning (ICML) (2009)
2. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
3. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
4. Chen, X., Mottaghi, R., Liu, X., Fidler, S., Urtasun, R., Yuille, A.: Detect what you can: Detecting and representing objects using holistic models and body parts. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
5. Chen, X., Yuille, A.L.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: Advances in Neural Information Processing Systems (NeurIPS) (2014)
6. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
7. Chen, Y., Rohrbach, M., Yan, Z., Shuicheng, Y., Feng, J., Kalantidis, Y.: Graph-based global reasoning networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
8. Chu, X., Ouyang, W., Wang, X., et al.: Crf-cnn: Modeling structured information in human pose estimation. In: Advances in Neural Information Processing Systems (NeurIPS) (2016)
9. Dhillon, I.S., Guan, Y., Kulis, B.: Weighted graph cuts without eigenvectors a multilevel approach. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2007)
10. Doering, A., Iqbal, U., Gall, J.: Joint flow: Temporal flow fields for multi person tracking. arXiv preprint arXiv:1805.04596 (2018)
11. Duvenaud, D.K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., Adams, R.P.: Convolutional networks on graphs for learning molecular fingerprints. In: Advances in Neural Information Processing Systems (NeurIPS) (2015)
12. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: Rmpe: Regional multi-person pose estimation. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
13. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. International Journal of Computer Vision (IJCV) (2005)
14. Fischler, M.A., Elschlager, R.A.: The representation and matching of pictorial structures. IEEE Transactions on Computers (1973)
15. Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: IEEE International Joint Conference on Neural Networks (IJCNN) (2005)
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. arXiv preprint arXiv:1703.06870 (2017)
17. Huang, S., Gong, M., Tao, D.: A coarse-fine network for keypoint localization. In: The IEEE International Conference on Computer Vision (ICCV) (2017)
18. Insafutdinov, E., Andriluka, M., Pishchulin, L., Tang, S., Levinkov, E., Andres, B., Schiele, B., Campus, S.I.: Arttrack: Articulated multi-person tracking in the wild. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

19. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In: European Conference on Computer Vision (ECCV) (2016)
20. Iqbal, U., Milan, A., Andriluka, M., Ensafutdinov, E., Pishchulin, L., Gall, J., B., S.: PoseTrack: A benchmark for human pose estimation and tracking. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
21. Iqbal, U., Milan, A., Gall, J.: Pose-track: Joint multi-person pose estimation and tracking. arXiv preprint arXiv:1611.07727 (2016)
22. Jin, S., Liu, W., Ouyang, W., Qian, C.: Multi-person articulated tracking with spatial and temporal embeddings. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
23. Jin, S., Ma, X., Han, Z., Wu, Y., Yang, W., Liu, W., Qian, C., Ouyang, W.: Towards multi-person pose tracking: Bottom-up and top-down methods. In: ICCV PoseTrack Workshop (2017)
24. Johnson, S., Everingham, M.: Clustered pose and nonlinear appearance models for human pose estimation. In: BMVC (2010)
25. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
26. Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H.S., Lu, C.: Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
27. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV) (2014)
28. Liu, W., Chen, J., Li, C., Qian, C., Chu, X., Hu, X.: A cascaded inception of inception network with attention modulated feature fusion for human pose estimation. In: The Thirty-Second AAAI Conference on Artificial Intelligence (2018)
29. Newell, A., Huang, Z., Deng, J.: Associative embedding: End-to-end learning for joint detection and grouping. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)
30. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision (ECCV) (2016)
31. Nie, X., Feng, J., Xing, J., Yan, S.: Generative partition networks for multi-person pose estimation. arXiv preprint arXiv:1705.07422 (2017)
32. Nie, X., Feng, J., Zhang, J., Yan, S.: Single-stage multi-person pose machines. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
33. Papandreou, G., Zhu, T., Chen, L.C., Gidaris, S., Tompson, J., Murphy, K.: Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. arXiv preprint arXiv:1803.08225 (2018)
34. Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., Murphy, K.: Towards accurate multi-person pose estimation in the wild. arXiv preprint arXiv:1701.01779 (2017)
35. Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P.V., Schiele, B.: Deepcut: Joint subset partition and labeling for multi person pose estimation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
36. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Transactions on Neural Networks (TNN) (2008)
37. Song, J., Andres, B., Black, M.J., Hilliges, O., Tang, S.: End-to-end learning for graph decomposition. In: The IEEE International Conference on Computer Vision (ICCV) (2019)

38. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. arXiv preprint arXiv:1902.09212 (2019)
39. 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)
40. Tian, Z., Chen, H., Shen, C.: Directpose: Direct end-to-end multi-person pose estimation. arXiv preprint arXiv:1911.07451 (2019)
41. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: Fully convolutional one-stage object detection. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
42. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems (NeurIPS) (2014)
43. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. International Conference on Learning Representations (ICLR) (2018)
44. Wang, J., Peng, Z., Lv, P., Sun, J., Zhou, B., Xu, M.: Bi-directional graph structure information model for multi-person pose estimation. arXiv preprint arXiv:1805.00603 (2018)
45. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (TOG) (2019)
46. Xiao, B., Wu, H., Wei, Y.: Simple baselines for human pose estimation and tracking. In: European Conference on Computer Vision (ECCV) (2018)
47. Xie, E., Sun, P., Song, X., Wang, W., Liu, X., Liang, D., Shen, C., Luo, P.: Polarmask: Single shot instance segmentation with polar representation. arXiv preprint arXiv:1909.13226 (2019)
48. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-Second AAAI Conference on Artificial Intelligence (AAAI) (2018)
49. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2012)
50. Zhang, H., Ouyang, H., Liu, S., Qi, X., Shen, X., Yang, R., Jia, J.: Human pose estimation with spatial contextual information. arXiv preprint arXiv:1901.01760 (2019)
51. Zhang, S.H., Li, R., Dong, X., Rosin, P., Cai, Z., Han, X., Yang, D., Huang, H., Hu, S.M.: Pose2seg: detection free human instance segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
52. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. arXiv preprint arXiv:1904.07850 (2019)