The Center of Attention: Center-Keypoint Grouping via Attention for Multi-Person Pose Estimation

Guillem Brasó   Nikita Kister   Laura Leal-Taixé
Technical University of Munich
{guillem.braso, n.kister, lealtaixe}@tum.de

Abstract

We introduce CenterGroup, an attention-based framework to estimate human poses from a set of identity-agnostic keypoints and person center predictions in an image. Our approach uses a transformer to obtain context-aware embeddings for all detected keypoints and centers and then applies multi-head attention to directly group joints into their corresponding person centers. While most bottom-up methods rely on non-learnable clustering at inference, CenterGroup uses a fully differentiable attention mechanism that we train end-to-end together with our keypoint detector. As a result, our method obtains state-of-the-art performance with up to 2.5x faster inference time than competing bottom-up approaches. Our code is available at https://github.com/dvl-tum/center-group

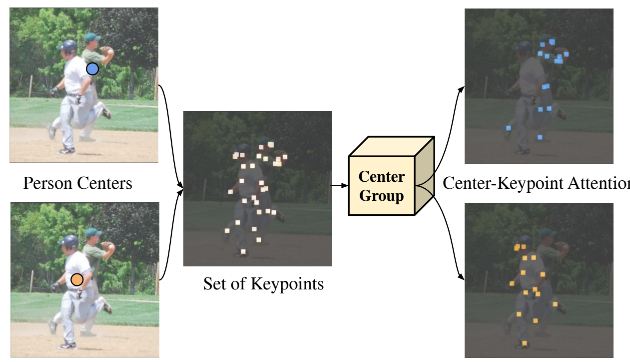

Figure 1: Given a set of predicted identity-agnostic keypoints and person centers, CenterGroup learns to assign keypoints into their corresponding centers with attention.

1. Introduction

Localizing the anatomical 2D keypoints of all humans in an image is a fundamental task in computer vision, with the ability to enable progress in applications such as virtual reality, human computer interaction, and human behavior analysis. It is also a common key component of algorithms for tasks such as action recognition [60, 13], multi-object tracking [21, 29], and generative models [8, 48].

Current methods typically follow one of two paradigms: *bottom-up* and *top-down*. Top-down approaches [10, 17, 18, 25, 31, 41, 59, 51, 52] divide the problem into two subtasks: (i) bounding box detection for all persons in the image, and (ii) joint localization for each person individually. Despite their success in some benchmarks [2, 1, 32], these two-step approaches lack efficiency due to their need to use a separate object detector, and their performance tends to degrade severely under heavy occlusions [31]. Bottom-up methods [6, 44, 26, 37, 40, 12, 28] follow a different approach, as they first detect identity-agnostic keypoints, and then group them into separate poses. Their lack of reliance on external object detectors and their ability to operate jointly over the entire set of keypoints in the image has allowed them to outperform top-down approaches in benchmarks where occlusions are common [31]. While recent work has significantly advanced the ability of bottom-up methods to accurately predict identity-free keypoints [12], current grouping algorithms still face significant drawbacks: since they generally rely on optimization algorithms, they cannot be trained end-to-end, and are often slow.

The keypoint grouping task can be formulated as a graph optimization problem in which nodes represent keypoints, and edge weights, which can be learned, represent their likelihood of belonging to the same human pose. Approaches ranging from integer linear programming [44, 26, 45], heuristics [37, 40, 6] or graph clustering [28] are then used to find the correct assignment. A common problem of bottom-up methods is that their learning objectives are poorly aligned with the real inference procedure: they learn affinities between keypoints but, at test time, grouping is performed by a separate algorithm which is not differentiable *per se*.

One-shot methods are an efficient alternative [39, 65, 58] to optimization-based bottom-up methods. Their general formulation consists in regressing a *root node* location per person, and then predicting offsets to keypoint locations. Since they are able to avoid the optimization-based grouping stage, they are significantly faster than their counterparts. However, given the inherent difficulty of predicting offsets under occlusions and scale variation, they are also significantly less accurate, and therefore have to rely on additional postprocessing techniques to obtain competitive performance [39, 65, 58].

We propose to tackle the limitations of current bottom-up grouping and one-shot algorithms with a novel framework based on attention. Instead of regressing offsets from a set of center nodes, our proposed CenterGroup uses attention to search for the best match between person centers and keypoints over the entire image. Our method retains the ability of bottom-up approaches to precisely predict keypoints from heatmaps, while maintaining the efficiency of one-shot methods. Furthermore, unlike standard bottom-up methods, CenterGroup does not require any test-time optimization and is end-to-end trainable.

More specifically, we first obtain proposals for person centers and identity-agnostic keypoints via heatmap regression. We then feed centers and keypoints to a transformer [54] to encode contextual information into their updated embeddings. Finally, the embeddings are used in a simple keypoint grouping scheme which maximizes the attention scores between person centers and keypoints belonging to the same pose. At test time, we extract poses by assigning to centers those keypoints with the corresponding highest attention score. Due to the simplicity of our grouping algorithm and the parallel nature of attention computation, CenterGroup is 2.5x faster than the current state-of-the-art bottom-up method [12], while having better performance.

Overall, we make the following **contributions**:

- We propose to tackle the pose estimation problem by grouping keypoints and person center predictions with a multi-head attention formulation that allows to train the model in an end-to-end fashion.

- We use a transformer to encode dependencies between bottom-up detected keypoints and centers to obtain context-enhanced embeddings, efficiently boosting the performance of our proposed grouping scheme.

- We achieve state-of-the-art results within an end-to-end framework that yields a speedup increase of up to 2.5x with respect to state-of-the-art [12].

2. Related Work

Top-down methods. Top-down methods [10, 17, 18, 25, 31, 41, 59, 51, 52, 56, 63, 22, 5, 38, 36, 50, 46, 53, 14] split the task into two steps. They first apply a person detector on the image and then perform single person pose estimator for each detected image region which is given by the bounding boxes. While being particularly strong at handling scale variaion, these methods struggle in cases of occlusion. To address these limitations, previous work has explored refining poses by exploiting the graph structure of the human skeletons with additional modules such as graph networks [56, 4, 46] or probabilistic graphical models[53]. While being more robust, these methods still rely on external detectors, and therefore cannot recover from missing boxes.

Bottom-up methods. Bottom-up methods [6, 44, 26, 37, 40, 12, 28, 30] start by detecting identity-free keypoints over the entire image. In a second step, a grouping algorithm assembles poses using pairwise similarity scores between keypoints. To predict these similarity scores, DeepCut and Person-Lab [44, 26, 40] predict offset fields that link joints belonging to the same person. Openpose and PifPaf [6, 30] predict part affinity fields which resemble human limbs and encode the position and orientation between pairs of keypoints. Associative embeddings [37] are a popular approach currently used by state-of-the-art [12]. They predict an embedding for every detected keypoint from convolutional features, and then use their pairwise euclidean distances as similarity scores. For all these methods, grouping is done by either graph partitioning [44, 26, 1, 45, 28, 27] or heuristic greedy parsing [37, 6, 40]. HGG[28] makes progress towards learning to group keypoints by using a graph neural network [47] on top of associative embeddings and training an edge and node classifier to hierarchically predict which keypoints belong together. While its graph network is trainable, it still relies on an external non-differentiable clustering algorithm [15] for grouping. CenterGroup does not need this clustering step and, instead, it uses attention as a form of learnable keypoint grouping.

One-shot methods. One-shot methods[39, 65, 58] avoid the grouping task by directly regressing keypoint locations from a set of predicted centers [39, 65] or anchors[58]. Both SPM[39] and CenterNet[65] regress offsets at center locations to regress each person's joints. In addition, [65] predicts keypoint heatmaps as standard bottom-up methods. It then combines both predictions by heuristically matching offsets to their closest predicted joints. While being more efficient than grouping-based approaches, this heuristic still does not perform on par with them, and suffers from the same problem of not being end-to-end learnable.

Transformers and attention. Transformers were initially introduced for machine translation, and became recently popular for computer vision tasks ranging from image classification [16, 9], object detection [7, 66], semantic segmentation [55, 57], video processing [62, 64], image generation [43], and hand pose estimation [23, 24]. They employ self-attention layers to model relations between entities in a global context. Their use for human pose estimation is still relatively unexplored: [49] employs transformer for human pose tracking, [35, 20] apply them to estimate 3D human poses and [61] use a transformer based architecture for explainable single person pose estimation.

3. Background: Multi-Head Attention

Our model uses multi-head attention and a transformer as main tools to perform grouping. Therefore, we start by providing a brief review of these techniques.

Multi-Head Attention (MHA) [54], the core component of the transformer model, aims at obtaining contextual representations from an unordered set of vectors by letting each vector *attend* over multiple representation subspaces of a (possibly different) set of vectors. More precisely, given a set of n $d-$dimensional *query* feature vectors, $Q := \{q_i \in \mathbb{R}^d\}_{i=1}^n$, and a set of m pairs of *key* and *value* vectors, $K := \{k_j \in \mathbb{R}^d\}_{j=1}^m$ and $V := \{v_j \in \mathbb{R}^d\}_{j=1}^m$ [1], MHA updates the query embeddings by linearly projecting the concatenation of h attention heads:

$$\text{MHA}(Q, K, V) = \text{concat}(\text{head}_1, \dots, \text{head}_h)W_O, \quad (1)$$

where $W_O \in \mathbb{R}^{(d_H * h)\times d}$ is a learnable matrix, and d_H is the dimensionality of each attention head. Each attention head computes, at every index $l \in \{1, \dots n\}$:

$$(head_i)_l = \sum_{j=1}^{m} \text{attn}_i(q_l, k_j) W_i^V v_j, \quad (2)$$

where the attention scores are computed as softmax-normalized[2] dot-products between keys and queries:

$$\text{attn}_i(q_l, k_j) := \frac{\exp((W_i^Q q_l)^T (W_i^K k_j))}{\sum_{t=1}^{m} \exp((W_i^Q q_l)^T (W_i^K k_t))} \quad (3)$$

where, $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d \times d_H}$ are learnable projection matrices.

Whenever these sets of key, query and values are the same, i.e., $Q = K = V$, one refers to *self-attention*, which is the core component of the transformer encoder architecture. Overall, transformer encoders are formed by stacking blocks of an initial layer of self-attention with skip-connection and layer normalization [3], followed by a feed-forward network and a second instance of layer normalization. For completeness, we provide a more detailed explanation of their architecture in the supplementary material.

4. Problem Formulation

We first provide an overview of the general formulation of our method and introduce notation.

[1]For notational simplicity, we assume queries, keys and values have the same dimensinality.

[2]Transformers use *scaled* dot product attention, which means that they normalize softmax outputs with the dimensionality of the projected embeddings, d_H. However, we omit this term as we will not use it in the remaining of the paper.

4.1. Problem Statement

Given an input image, we aim to obtain the set of poses $\mathcal{P}$ corresponding to all persons in the image. Let J be the number of joints being considered. Each pose can be uniquely determined by the 2D location and visibility of its J joints. Formally, for every pose $p \in \mathcal{P}$, we refer to its joint locations as $\text{loc}_p^i \in \mathbb{R}^2$ for every $i \in \{1, \dots, J\}$. We denote the visibility of each joint of pose p as $\text{vis}_p^i \in \{0, 1\}$, and assign it 1 whenever the joint is visible, and 0 otherwise.

Our approach operates over a set of predicted identity-agnostic joint keypoints in the image, which we refer to as $\mathcal{K}$. Each keypoint $k \in \mathcal{K}$ can be identified by its 2D location $\text{loc}_k \in \mathbb{R}$ and its predicted type $\text{type}_k \in \{1, \dots, J\}$. Our method also predicts an additional set of targets corresponding to the center locations of persons in the image. We denote as $\mathcal{C}$ the set of detected person centers.

4.2. Grouping Keypoints and Centers

Standard bottom-up methods learn a similarity score $\text{sim}(k_1, k_2)$ for every pair of detected keypoints $k_1, k_2 \in \mathcal{K}$, and use them to form poses by clustering keypoints that are most similar. One-shot methods, instead, directly use predicted person centers and regress displacement offsets from centers to joint locations to avoid expensive grouping.

Inspired by these approaches, we propose to perform human pose estimation by learning similarity score $\text{sim}_i(c, k) \in \mathbb{R}^+$ between every pair of detected person center $c \in \mathcal{C}$ and keypoints $k \in \mathcal{K}$ and type $i \in \{1, \dots, J\}$. By being able to estimate the similarity between center nodes and keypoints, as opposed to the similarity between pairs of keypoints, we are able to reduce the complexity of the grouping task significantly. Instead of requiring a graph clustering algorithm, we formulate the grouping task as a simple nearest-neighbor search problem. Namely, for every predicted center $c \in \mathcal{C}$, we obtain its corresponding pose by retrieving the locations of its most similar detected keypoint $k^* \in \mathcal{K}$ of a target type $i \in \{1, \dots, J\}$ according to sim_i. Formally, the predicted location $\widehat{\text{loc}}_c^i$ of joint type i for center c can be obtained as $\widehat{\text{loc}}_c^i = \text{loc}_{k^*}$, where

$$k^* = \arg\max_{k \in \mathcal{K}} \text{sim}_i(c, k) \quad (4)$$

Since our approach operates directly over the set of detected keypoint locations, there is no need for additional postprocessing to obtain precise joint locations, unlike offset-based methods [65, 39].

4.3. Attention as Differentiable Keypoint Selection

The main drawback from the aforementioned procedure is that it is not end-to-end trainable, as it involves an $\arg\max$ operation over the detected keypoints. We circumvent this issue by formulating the nearest neighbor search

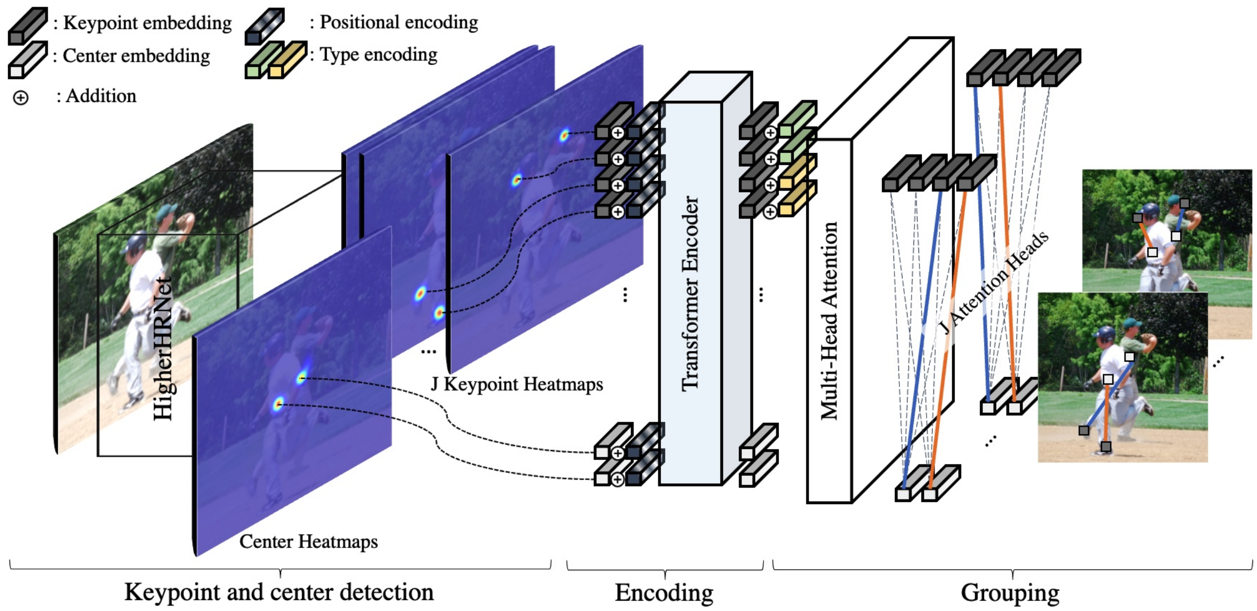

Figure 2: Our method receives a single RGB image as input and predicts a set of identity-agnostic keypoints and person centers with a HigherHRNet[12] by heatmap regression. It then extracts features from our CNN's last layer, augments them with positional encodings and feeds them to a transformer encoder that returns context-aware embeddings for every keypoint and center. These embeddings are then fed to an attention module that predicts which joints correspond to each center.

task as a differentiable attention mechanism. To do so, we treat person centers as our set of queries, and keypoints as our set of keys, and obtain their similarity scores by computing their dot-product in a learned embedding space for every joint type $i \in \{1, \ldots, J\}$. We then normalize the scores with a softmax operator to replace the non-differentiable $\arg\max$. The resulting coefficients are used during training to directly predict keypoint locations for every joint type $i \in \{1, \ldots, J\}$ and every person center c as:

$$\widehat{\text{loc}}_c^i := \sum_{k \in \mathcal{K}} \text{attn}_i(c, k)\text{loc}_k \tag{5}$$

where loc_k are the coordinates of the detected keypoint k.

Predictions resulting from equation 5 can then be minimized by directly computing their $L1$ loss with respect to the ground truth location. Note that since the detected keypoint coordinates are fixed in Equation 5, in order to minimize the loss our network needs to assign the highest attention score to the keypoints which location is closest to the ground truth coordinates loc_c^i. Moreover, in the limit, when $\max_{k \in \mathcal{K}} \text{attn}_i(c, k) = 1$, Equation 5 becomes equivalent to just computing a standard $\arg\max$. Therefore, the attention coefficients act as a differentiable mechanism for selecting keypoints from person centers based on their dot-product similarity. This procedure still allows us to use the simple $\arg\max$ operator at test-time to efficiently retrieve keypoints from centers as in Equation 4.

5. Method

We exploit the formulation described in the previous section within an end-to-end bottom-up pipeline for pose estimation. In this section, we first provide a general overview of it, and then explain each of its components in detail.

5.1. Overview

Our method, CenterGroup, consists of three main stages, which are summarized in Figure 2:

1. **Keypoint and center detection.** The location of identity-agnostic keypoints and person centers is obtained by heatmap regression following HigherHRNet [12]. The output is a variable number of high-scoring joint and person center detections.

2. **Encoding keypoints and centers.** For every detected keypoint and center, we extract features from a CNN backbone, and augment them with additional embeddings encoding their spatial position. These embeddings are fed to a transformer [54], yielding updated embeddings with enhanced contextual information.

3. **Keypoint grouping.** We use the embeddings obtained from the previous stage and compute dot-product attention scores between person centers and keypoints, and normalize them in order to obtain a soft-assignment between persons and keypoints. Additionally, we use the transformer embeddings to classify center nodes into true and false positives, and determine the visibility of each keypoint.

5.2. Keypoint and Center Detection

In the first stage of our pipeline, we start by detecting identity agnostic-keypoints and person centers.

Heatmap regression. We follow HigherHRNet [12] to obtain identity-agnostic keypoint proposals for each of J joint types being considered. HigherHRNet uses an HRNet [51] backbone, followed by two keypoint prediction heads that regress heatmaps at 1/4 and 1/2 of the original image scale for every joint type. Heatmaps are trained to follow a gaussian distribution centered at ground truth keypoint locations. During training, both heatmaps are supervised independently with a minimum-squared error loss. At inference, heatmaps are upsampled and aggregated to obtain a single heatmap at full image resolution.

Person centers. In addition to joints, we regress a new heatmap corresponding to person centers, also at resolutions 1/4 and 1/2. Following [39], given a ground truth pose $p \in \mathcal{P}$ with joint locations $\{\text{loc}_p^i \in \mathbb{R}^2\}_{i=1}^J$, the location of its center is computed as the center of mass of the visible joints, i.e.,

$$\text{loc}_p := \frac{1}{N_p} \sum_{\text{vis}_p^i=1} \text{loc}_p^i. \tag{6}$$

where $N_p := \sum_{i=1}^J \text{vis}_p^i$ is the number of visible joints in pose p. Note that we identify the pose location as that of its center, and hence write loc_p.

5.3. Encoding Keypoints and Centers

Given the set of predicted keypoints and centers from the first stage, our goal is to obtain discriminative embeddings encoding contextual information. These embeddings will then be used in our grouping module in order to predict associations among keypoints and centers. Hence, it is desirable for them to encode global context.Towards this end, we use a transformer encoder that yields updated embeddings for every keypoint and center.

Initial features. We add one additional residual block [19] to our backbone's last feature map at 1/4 of the original resolution. For every detected keypoint and center, we obtain an initial embedding vector by extracting the vector at its corresponding location from the resulting feature map, and feed it to a two-layer Multi-Layer Perceptron (MLP) to project it onto a higher dimensionality.

Positional encodings. CNN features struggle to encode the position of different keypoints [34]. However, spatial information offers an important cue for keypoint grouping. Hence, similarly to previous work [7, 66, 16, 57], we use fixed sinusoidal features encoding the absolute x and y axis locations at different frequencies. As a result, we obtain a new vector of dimensionality d and sum it element-wise to the initial features of every detected keypoint and center.

Transformer encoder. In order to encode global context among every detected person and keypoint, we take their initial features, augmented with positional encodings, and feed them to a transformer encoder. As a result, we obtain updated embeddings h_k and h_c for every detected keypoint $k \in \mathcal{K}$ and center $c \in \mathcal{C}$. Our transformer architecture follows the one described in Section 3, and is described in detail in the supplementary material.

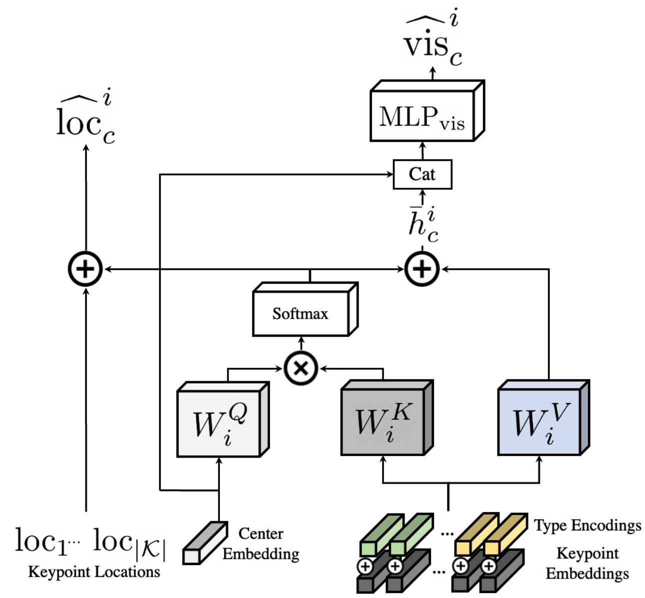

Figure 3: Overview of our Grouping Module. Attention between keypoint and center embeddings is used in order to predict joint locations and visibility for a given center.

5.4. Keypoint Grouping

In the last stage of our pipeline, we construct poses by using the embeddings produced by our transformer to determine which keypoints belong to which person centers via pairwise attention scores. As explained in Section 4.3, we use attention as a differentiable approximation of keypoint selection from centers. In addition, we predict two additional targets for every center: the visibility of each of their keypoints, and the probability that they represent a true pose. This module is summarized in Figure 3.

Classifying centers. We start by identifying which predicted centers locations correspond to the ground truth poses by matching them based on their locations[3]. As a result, each predicted center is labelled with a binary target y_{center}^c, set to 1 if the center is matched and 0 otherwise. We then use a small multi-layer perceptron to classify the center embeddings produced by our transformer, h_c, and supervise the resulting prediction $\text{MLP}_{\text{center}}(h_c)$ with a focal loss [33].

Predicting joint locations. For every predicted center $c \in \mathcal{C}$ such that $y_{\text{center}}^c = 1$, we aim to predict the 2D coordi-

[3] More details are provided in the supplementary material.

nates of each of its joints $\widehat{\text{loc}}_c^i$ of every type $i \in \{1, \dots, J\}$. For each joint type $i \in \{1, \dots, J\}$, we define a pair of learnable projection matrices W_i^K and W_i^Q, and a learned *type encoding* vector $\phi_i \in \mathbb{R}^d$. The goal of the projection matrices is to map the center embeddings h_c and keypoint embeddings h_k, into a discriminative representation in which their dot-product will encode their likelihood being a good match for type i. We compute their similarity as:

$$\text{sim}_i(c, k) = (W_i^Q h_c)^T W_i^K (h_k + \phi_{\text{type}_k}) \tag{7}$$

Note that the learned embedding ϕ_{type_k} is added to the keypoint embedding h_k before the multiplication. Its goal is to encode the initial type predicted by our keypoint detector for keypoint k. Intuitively, when searching for joints of a target type i, it is desirable for our network to still be able to consider joints of all predicted types in order to recover from type errors made by the keypoint detector. For instance, for a target type i, such as left ankle, some predicted types by the detector (e.g., right ankle) are more likely to be better matching candidates than others (e.g. nose). By using the learnable encoding ϕ_{type_k} for each keypoint before computing the projected embedding with W_i^K, we allow our network to explicitely account for the relationships between the target type i, and the predicted type of k, type_k, in a learnable manner.

With the similarity scores from Equation 7, the final attention scores are computed by normalizing them with a softmax operation over the entire set of keypoints:

$$\text{attn}_i(c, k) = \frac{\exp\left(\text{sim}_i(c, k)\right)}{\sum_{\bar{k} \in \mathcal{K}} \exp(\text{sim}_i(c, \bar{k}))} \tag{8}$$

Finally, we obtain the predicted locations as in Eq. 5:

$$\widehat{\text{loc}}_c^i = \sum_{k \in \mathcal{K}} \text{attn}_i(c, k) \text{loc}_k, \tag{9}$$

and supervise them with an $L1$ loss as:

$$\mathcal{L}_{\text{loc}} = \sum_{c \in \mathcal{C} | y_{\text{center}}^c = 1} \sum_{vis_c^i = 1} |\widehat{\text{loc}}_c^i - \text{loc}_c^i|, \tag{10}$$

where the learnable ground locations of every joint for center loc_c^i are those of its matched ground truth pose.

Overall, this procedure can be interpreted as an instance of an attention head in which center and keypoint embeddings are *queries* and *keys*, respectively, and keypoint locations act as *values*. Note that we use different matrices W_i^K and W_i^Q for every target type $i \in \{1, \dots, J\}$, which is equivalent to having J different heads.

Predicting keypoint visibility. One drawback of the attention mechanism we have described is that, due to the softmax normalization, it may still predict high attention scores between centers and keypoints for a target type i even if a given center has no corresponding visible keypoint of that type in the image. We address this problem by exploiting the attention mechanism to explicitly classify whether the predicted keypoints are visible. To do so, we introduce an additional projection matrix for every head W_i^V, and reuse the type encodings and attention scores already computed to predict locations to compute a weighted aggregation analogous to that in Equation 10:

$$\bar{h}_c^i = \sum_{k \in \mathcal{K}} \text{attn}(c, k) W_i^V (h_k + \phi_{\text{type}_k}) \tag{11}$$

we then concatenate $\bar{h}_c^i$ and h_c, and classify the resulting vector with an additional multilayer perceptron, MLP_{vis} as either visible or not visible. We supervise the result with a focal loss[4]. Intuitively, whenever keypoints are not visible, the original embeddings h_k and h_c will not be aligned, and therefore, neither will be $\bar{h}_c$ and h_c. Hence, their concatenation can be discriminatively used to identify when a joint has no good keypoint candidate for the target joint type.

6. Experiments

In this section, we detail the experimental evaluation of our method. We divide it into ablation studies and comparison to state-of-the-art on two large-scale public datasets. For implementation details, we refer the reader to the supplementary material.

6.1. Datasets and Evaluation Metrics

COCO Keypoint Detection. The COCO dataset [32] is a large-scale benchmark containing large variety of everyday life situations. It contains over 200,000 images and 17 keypoints annotations for more than 250,000 human instances, which are split in approximately 150,000, 80,000 and 20,000 instances are for training, testing and, validation, respectively. We train our models on the *train2017* split only, perform our ablation studies on *val2017*, and report our final results on the *test-dev2017* split.

CrowdPose. The CrowdPose dataset [31] is a challenging benchmark with the goal to evaluate the robustness of methods in crowded scenes. Unlike COCO, in which the majority of images contain few instances, the *crowd index* in CrowdPose follows a uniform distribution [31]. The dataset contains a total of 20,000 images and a total of 80,000 instances annotated with 14 keypoints. Images are split in a ratio 5:4:1 for training, validation, and testing. Following [12], we train our model on the train and validation splits combined, and report the final performance on the test set.

[4]This loss is only computed whenever the given joint in the predicted center is labelled as not visible in the ground truth, or the predicted keypoint has is has small euclidean distance with respect to the with the ground truth keypoint.

#	Method	Group.	Type Agnostic	Type Encoding	Transformer	Pos. Encoding	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
1	Offsets + keypoint match.[65]						65.3	86.4	71.4	59.1	75.0
2	AE [12, 37]						67.1	86.2	73.0	61.5	76.1
3	Ours w/o K&C transformer enc.	✓					67.5	86.7	72.7	62.0	76.6
4	Ours w/o K&C transformer enc.	✓	✓				67.5	86.8	72.9	60.8	77.3
5	Ours w/o K&C transformer enc.	✓	✓	✓			67.9	87.4	73.2	61.4	77.4
6	Ours	✓	✓	✓	✓		68.4	87.5	73.9	62.0	77.6
7	Ours	✓	✓	✓	✓	✓	**68.6**	**87.6**	**74.1**	**62.0**	**78.0**

Table 1: Ablation study on the **COCO2017 val** split.

Evaluation metrics. The aforementioned datasets use average precision (AP) as their main metric. AP computation is based on the Object Keypoint Similarity (OKS) [32] score among detected and ground truth poses. AP is the result of average precision scores for OKS thresholds $0.50, 0.55..., 0.90, 0.95$. We also report AP for thresholds 0.5 and 0.75, namely, AP^{50} and AP^{75}. In addition, for COCO we report AP^{L} and AP^{M}, which corresponds to AP over medium and large-sized instances respectively. For CrowdPose, we also report AP^{E}, AP^{M}, AP^{H}, which stands for AP scores over easy, medium and hard instances, according to dataset annotations.

6.2. Ablation Study

To determine the individual contribution of each of our model's main components, we perform an ablation study on the COCO val2017 split, with HRNet32 backbone and input size 512x512. All results are reported with flip-testing, following [12, 37, 28], and without top-down refinement.

Baselines. CenterGroup can be naturally compared to two alternative frameworks. First, associative embeddings [37] (Tab. 1, row #1), since they are the method used originally by our keypoint detection network [12]. Second, one-shot or offset-based methods [65, 39], which also use person center predictions, but use offset regression to obtain the final results. For a fair comparison, we reimplement [65] with our HigherHRNet backbone and report its performance for its strongest variant, which predicts keypoint heatmaps, center heatmaps and center offsets, and matches centers to their closest predicted keypoint (Tab. 1, row #2).

Grouping Module. We consider our model without the transformer encoder to isolate the effect of our Grouping Module. We compare three versions of it. In the first one, the attention head corresponding to the prediction of each keypoint type is only allowed to attend over keypoints of the same type that our keypoint detection network detects, and therefore is not able to overcome joint type mistakes made by the detector. This setting (Tab. 1, row #3) already outperforms our baselines, which confirms the superiority of CenterGroup over AE-based grouping and offset-based methods. In rows #4 and #5, we allow each head to attend over keypoints from the entire set of predicted heatmaps, and refer to them as *type-agnostic*. In row #5, we further use type encoding in the attention computation, as explained in Section 5.4, and observe that they significantly improve upon type-agnostic grouping.

Feature encoding. In rows #6 and #7 of Table 1, we further analyze the effect of using the keypoint and center transformer encoding before the routing module. This yields a significant performance boost, which confirms the importance of encoding long-range interactions between keypoints. Further enhancing the initial embeddings with positional encodings allows the transformer to explicitly use spatial information and gives up to 0.4 AP points of improvement for large persons.

Loss terms. We also assess the importance of our additional center and visibility classification losses, the results can be found in Table 2. Without them, we score our predicted poses by directly assigning them the confidence of its predicted center from heatmaps. We observe that replacing the heatmap score with the classification score obtained from our transformer's embeddings (row #2) already provides a significant boost. We then experiment with either using a simple MLP over those features to predict the visibility of every keypoint (row #3), compared to using our attention-based model shown in Figure 3, results in row #4. We observe that both yield a significant improvement, but our attention-based model performs best.

Runtime analysis. In Table 3, we report the overall speed of our method when compared to our baselines. All models are run on the same machine with a single NVIDIA RTX5000 GPU, with batch size 1 and flip testing. We report: grouping runtime, i.e., all computations after keypoint detection, which in our case includes the transformer and grouping attention forward pass, and the overall runtime, which always adds 126ms corresponding to HigherHRNet. The overall runtime of CenterGroup is similar to [65], while we get significantly better results. Compared to AE-based grouping, our keypoint attention grouping is over 6x faster.

6.3. Benchmark Evaluation

COCO Keypoint Detection. In Table 4, we compare CenterGroup against state-of-the-art methods on the COCO dataset. Our method achieves the best performance among

#	Class. Cent.	Vis. w/ MLP	Vis. w/ Attn.	AP	AP^{M}	AP^{L}
1				66.5	61.4	75.5
2	✓			67.1	61.0	76.2
3	✓	✓		68.2	61.8	77.5
4	✓		✓	**68.6**	**62.0**	**78.0**

Table 2: Ablation study on loss terms.

Method	Group. Time (ms)	Time (ms)	AP	AP^{M}	AP^{L}
Offsets + match[65]	**20**	**146**	65.3	59.1	75.0
AE[37]	327	453	67.1	60.7	76.0
Ours	52	178	**68.6**	**62.0**	**78.0**

Table 3: Runtime analysis of different grouping methods.

Method	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}
Top-down methods					
Mask-RCNN [18]	63.1	87.3	68.7	57.8	71.4
G-RMI [42]	64.0	85.5	71.3	62.3	70.0
Integral Pose Regression [52]	67.8	88.2	74.8	63.9	74.0
CPN [11]	72.1	91.4	80.0	68.7	77.2
RMPE [17]	72.3	86.1	79.1	68.0	78.6
CFN [25]	72.6	86.1	69.7	**78.3**	64.1
SimpleBaseline [59]	73.7	91.9	81.1	70.3	80.0
HRNet-W48 [51]	**75.5**	**92.5**	**83.3**	71.9	**81.5**
Bottom-up methods					
OpenPose* [6]	61.8	84.9	67.5	57.1	68.2
Hourglass*+ [37]	65.5	86.8	72.3	60.6	72.6
PifPaf[30]	66.7	-	-	62.4	72.9
SPM*+ [39]	66.9	88.5	72.9	62.6	73.1
HGG+ [28]	67.6	85.1	73.7	62.7	74.6
PersonLab+ [40]	68.7	89.0	75.4	66.6	75.8
HrHRNet-W32[12]	66.4	87.5	72.8	61.2	74.2
HrHRNet-W48[12]	68.4	88.2	75.1	64.4	74.2
HrHRNet-W48+ [12]	70.5	89.3	77.2	66.6	75.8
Ours w/ HrHRNet-W32	67.6	88.7	73.6	61.9	75.6
Ours w/ HrHRNet-W48	69.6	89.7	76.0	64.9	76.3
Ours w/ HrHRNet-W32+	70.0	89.9	76.6	65.2	**77.1**
Ours w/ HrHRNet-W48+	**71.1**	**90.5**	**77.5**	**66.9**	76.7

Table 4: Comparisons with state of the art methods on the **COCO2017 test-dev** split. * means top-down refinement, and + means multi-scale testing.

all bottom-up methods, for both single and multi-scale testing, and outperforms HigherHRNet, which uses AE-based Grouping [37], by 1.2 AP points in the single-scale setting. We observe that our achievements are most significant in AP^{L}. This can be explained by the ability of our attention module to capture long-range interactions between joints that are far apart. Overall, strong results in COCO, combined with our faster inference speed, show that CenterGroup is a more efficient alternative to current bottom-up methods [65]. We provide additional analysis in the supplementary material.

CrowdPose. In Table 5, we show the test-set results for our model trained on CrowdPose. Unlike COCO, where

Method	AP	AP^{50}	AP^{75}	AP^{E}	AP^{M}	AP^{H}
Top-down methods						
Mask-RCNN [18]	57.2	83.5	60.3	69.4	57.9	45.8
SimpleBaseline [59]	60.8	81.4	65.7	71.4	61.2	51.2
AlphaPose [17]	61.0	81.3	66.0	71.2	61.4	51.1
Top-down with refinement						
SPPE [31]	66.0	84.2	71.5	75.5	66.3	57.4
Bottom-up methods						
OpenPose*[6]	-	-	-	62.7	48.7	32.3
HrHRNet-W48[12]	65.9	86.4	70.6	73.3	66.5	57.9
HrHRNet-W48+ [12]	67.6	87.4	72.6	75.8	68.1	58.9
Ours w/ HrHRNet-W48	67.6	87.7	72.7	73.9	68.2	60.3
Ours w/ HrHRNet-W48+	**69.4**	**88.6**	**74.6**	**76.6**	**70.0**	**61.5**

Table 5: Comparisions with state of the art methods on the **CrowdPose test** set. Superscripts E, M, H mean easy, medium and hard.+ means multi-scale test, and * means top-down refinement.

top-down methods show superior performance, bottom-up methods outperform their top-down counterparts in CrowdPose, since this dataset is focused on much more challenging images with severe occlusions. In this setting, CenterGroup shows its full potential and obtains state-of-the-art performance among all methods by 1.8 AP points. Most importantly, our improvement is most significant in the hard regime (AP^{H}), where we improve upon state-of-the-art by 2.4 and 2.6 AP points for single and multi-scale testing, respectively.

This proves that our end-to-end learnable formulation does benefit from being trained on a dataset in which occlusions are common, and results in better generalization to new, challenging images. Overall, we show that our end-to-end trainable method can outperform top-down and bottom-up approaches on difficult scenarios with severe occlusions, where reasoning about keypoint detection and grouping jointly has a clear benefit.

7. Conclusion

We have proposed an end-to-end attention-based framework for bottom-up human pose estimation. We have demonstrated that CenterGroup has better performance than existing state-of-the-art methods, particularly in crowded images, while being significantly more efficient. We hope that our approach will inspire future work to explore the potential of attention mechanisms, as well as general learning-based alternatives to optimization-based grouping for bottom-up human pose estimation.

Acknowledgments. This project was partially funded by the Sofja Kovalevskaja Award of the Humboldt Foundation and by the German Federal Ministry of Education and Research (BMBF) under Grant No. 01IS18036B. The authors of this work take full responsibility for its content.

References

[1] M. Andriluka, U. Iqbal, E. Ensafutdinov, L. Pishchulin, A. Milan, J. Gall, and Schiele B. PoseTrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018. 1, 2

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 1

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. In *arXiv preprint arXiv:1706.03762*, 2016. 3

[4] Adrian Bulat and Georgios Tzimiropoulos. Human pose estimation via convolutional part heatmap regression. In *European Conference on Computer Vision*, pages 717–732. Springer, 2016. 2

[5] Yuanhao Cai, Zhicheng Wang, Zhengxiong Luo, Binyi Yin, Angang Du, Haoqian Wang, Xiangyu Zhang, Xinyu Zhou, Erjin Zhou, and Jian Sun. Learning delicate local representations for multi-person pose estimation. In *European Conference on Computer Vision*, pages 455–472. Springer, 2020. 2

[6] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 1, 2, 8

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2, 5

[8] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. Everybody dance now. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 1

[9] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International Conference on Machine Learning*, pages 1691–1703. PMLR, 2020. 2

[10] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018. 1, 2

[11] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7103–7112, 2018. 8

[12] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S. Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2, 4, 5, 6, 7, 8

[13] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[14] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, 2017. 2

[15] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007. 2

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 5

[17] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017. 1, 2, 8

[18] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 1, 2, 8

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5

[20] Yihui He, Rui Yan, Katerina Fragkiadaki, and Shoou-I Yu. Epipolar transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7779–7788, 2020. 2

[21] R. Henschel, Y. Zou, and B. Rosenhahn. Multiple people tracking using body and joint detections. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 770–779, 2019. 1

[22] Junjie Huang, Zheng Zhu, Feng Guo, and Guan Huang. The devil is in the details: Delving into unbiased data processing for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5700–5709, 2020. 2

[23] Lin Huang, Jianchao Tan, Ji Liu, and Junsong Yuan. Handtransformer: Non-autoregressive structured modeling for 3d hand pose estimation. In *European Conference on Computer Vision*, pages 17–33. Springer, 2020. 2

[24] Lin Huang, Jianchao Tan, Jingjing Meng, Ji Liu, and Junsong Yuan. Hot-net: Non-autoregressive transformer for 3d hand-object pose estimation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 3136–3145, 2020. 2

[25] Shaoli Huang, Mingming Gong, and Dacheng Tao. A coarse-fine network for keypoint localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3028–3037, 2017. 1, 2, 8

[26] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation

model. In *European Conference on Computer Vision*, pages 34–50. Springer, 2016. 1, 2

[27] Umar Iqbal and Juergen Gall. Multi-person pose estimation with local joint-to-person associations. In *European Conference on Computer Vision*, pages 627–642. Springer, 2016. 2

[28] Sheng Jin, Wentao Liu, Enze Xie, Wenhai Wang, Chen Qian, Wanli Ouyang, and Ping Luo. Differentiable hierarchical graph grouping for multi-person pose estimation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 718–734, Cham, 2020. Springer International Publishing. 1, 2, 7, 8

[29] Margret Keuper, Siyu Tang, Bjorn Andres, Thomas Brox, and Bernt Schiele. Motion segmentation multiple object tracking by correlation co-clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 10 2018. 1

[30] Sven Kreiss, Lorenzo Bertoni, and Alexandre Alahi. Pifpaf: Composite fields for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11977–11986, 2019. 2, 8

[31] Jiefeng Li, Can Wang, Hao Zhu, Yihuan Mao, Hao-Shu Fang, and Cewu Lu. Crowdpose: Efficient crowded scenes pose estimation and a new benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2, 6, 8

[32] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 1, 6, 7

[33] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 5

[34] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in Neural Information Processing Systems*, 2018. 5

[35] Adrian Llopart. Liftformer: 3d human pose estimation using attention models. *arXiv preprint arXiv:2009.00348*, 2020. 2

[36] Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Posefix: Model-agnostic general human pose refinement network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7773–7781, 2019. 2

[37] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 2277–2287. Curran Associates, Inc., 2017. 1, 2, 7, 8

[38] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016. 2

[39] Xuecheng Nie, Jiashi Feng, Jianfeng Zhang, and Shuicheng Yan. Single-stage multi-person pose machines. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6951–6960, 2019. 1, 2, 3, 5, 7, 8

[40] George Papandreou, Tyler Zhu, Liang-Chieh Chen, Spyros Gidaris, Jonathan Tompson, and Kevin Murphy. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 269–286, 2018. 1, 2, 8

[41] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017. 1, 2

[42] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017. 8

[43] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018. 2

[44] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4929–4937, 2016. 1, 2

[45] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele. Articulated multi-person tracking in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 1, 2

[46] Lingteng Qiu, Xuanye Zhang, Yanran Li, Guanbin Li, Xiaojun Wu, Zixiang Xiong, Xiaoguang Han, and Shuguang Cui. Peeking into occluded joints: A novel framework for crowd pose estimation. In *European Conference on Computer Vision*, pages 488–504. Springer, 2020. 2

[47] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. 2

[48] Aliaksandr Siarohin, Stéphane Lathuilière, Enver Sangineto, and Nicu Sebe. Appearance and pose-conditioned human image generation using deformable gans, 2019. 1

[49] Michael Snower, Asim Kadav, Farley Lai, and Hans Peter Graf. 15 keypoints is all you need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6738–6748, 2020. 2

[50] Kai Su, Dongdong Yu, Zhenqi Xu, Xin Geng, and Changhu Wang. Multi-person pose estimation with enhanced channel-wise and spatial information. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5674–5682, 2019. 2

[51] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5693–5703, 2019. 1, 2, 5, 8

[52] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 529–545, 2018. 1, 2, 8

[53] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 190–206, 2018. 2

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 2, 3, 4

[55] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. *arXiv preprint arXiv:2012.00759*, 2020. 2

[56] Jian Wang, Xiang Long, Yuan Gao, Errui Ding, and Shilei Wen. Graph-pcnn: Two stage human pose estimation with graph pose refinement. In *European Conference on Computer Vision*, pages 492–508. Springer, 2020. 2

[57] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, 2020. 2, 5

[58] Fangyun Wei, Xiao Sun, Hongyang Li, Jingdong Wang, and Stephen Lin. Point-set anchors for object detection, instance segmentation and pose estimation. In *European Conference on Computer Vision*, pages 527–544. Springer, 2020. 1, 2

[59] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *Proceedings of the European conference on computer vision (ECCV)*, pages 466–481, 2018. 1, 2, 8

[60] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition, 2018. 1

[61] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Towards explainable human pose estimation by transformer. *arXiv preprint arXiv:2012.14214*, 2020. 2

[62] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *European Conference on Computer Vision*, pages 528–543. Springer, 2020. 2

[63] Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7093–7102, 2020. 2

[64] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018. 2

[65] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 1, 2, 3, 7, 8

[66] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 5