

# Rethinking pose estimation in crowds: overcoming the detection information bottleneck and ambiguity

Mu Zhou\*    Lucas Stoffl\*    Mackenzie Weygandt Mathis    Alexander Mathis  
École Polytechnique Fédérale de Lausanne (EPFL)

alexander.mathis@epfl.ch

## Abstract

Frequent interactions between individuals are a fundamental challenge for pose estimation algorithms. Current pipelines either use an object detector together with a pose estimator (top-down approach), or localize all body parts first and then link them to predict the pose of individuals (bottom-up). Yet, when individuals closely interact, top-down methods are ill-defined due to overlapping individuals, and bottom-up methods often falsely infer connections to distant bodyparts. Thus, we propose a novel pipeline called bottom-up conditioned top-down pose estimation (BUCTD) that combines the strengths of bottom-up and top-down methods. Specifically, we propose to use a bottom-up model as the detector, which in addition to an estimated bounding box provides a pose proposal that is fed as condition to an attention-based top-down model. We demonstrate the performance and efficiency of our approach on animal and human pose estimation benchmarks. On CrowdPose and OCHuman, we outperform previous state-of-the-art models by a significant margin. We achieve 78.5 AP on CrowdPose and 48.5 AP on OCHuman, an improvement of 8.6% and 7.8% over the prior art, respectively. Furthermore, we show that our method strongly improves the performance on multi-animal benchmarks involving fish and monkeys. The code is available at <https://github.com/amathislab/BUCTD>

## 1. Introduction

Imagine somebody hands you an image of a person and asks you “to annotate the pose”. For your exquisite primate visual system this is a trivial task that you can readily achieve. Now imagine somebody hands you another image that contains two people, arm-in-arm. You are likely frustrated and will ask whose pose you should annotate? In response to whose pose you should annotate, your opponent will likely point at the person she has in mind. Based on the pointing, it’s again easy to annotate the right pose. Our

\*Authors contributed equally to this work.

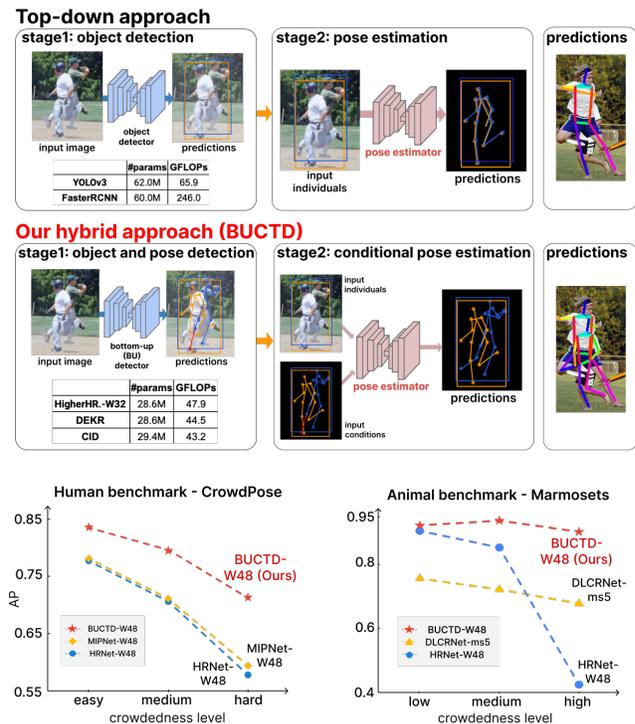


Figure 1. **Overview of our bottom-up conditioned top-down pose estimation (BUCTD) approach and benchmarking results.** BUCTD uses a bottom-up pose model as instance detector, which is computationally cheaper than existing, widely-used object detectors (see inset Table). The pose proposals from the pose detector are used to calculate bounding boxes and to condition our novel, conditional top-down stage. Note that, as in a standard top-down paradigm, only one image crop plus its corresponding conditional pose is presented to the BUCTD. We can substantially boost performance on both human & animal benchmarks, with especially large gains in crowded scenes (Tables 2, 7).

work proposes a hybrid deep learning framework for pose estimation that is inspired by this interaction.

This simple interaction highlights the ambiguity problem of top-down approaches in crowds. They first localize individuals with a dedicated object detector [33, 34, 12, 27]

and then perform single-instance pose estimation [13, 8, 39, 23, 42, 25, 40]. In contrast, bottom-up approaches first localize all body parts in the image and then assemble them into poses of each of the individuals simultaneously [19, 6, 31, 9, 16, 22, 36]. Yet, when individuals closely interact, top-down methods are ill-defined as it is unclear which pose should be predicted within a bounding box that contains multiple individuals. Therefore, occluded individuals will often be ignored by top-down methods (Figure 1). In contrast, as bottom-up approaches reason over the complete scene they may not have this problem. Bottom-up approaches can localize all individuals, but often struggle to make accurate predictions.

To overcome those limitations we propose a simple yet effective framework called **Bottom-Up Conditioned Top-Down** pose estimation (BUCTD). Our solution, is inspired by the interaction that we described. Instead of using object detectors, we propose to use bottom-up pose estimation models as detectors. The output poses are used to estimate bounding boxes of the individuals, and also serve as a “pointing” mechanism, that indicates whose pose should be predicted. To also process the “pointing” input, we generalize top-down models to conditional-top down (CTD) models, which present the second stage of our BUCTD framework. CTD models take a cropped image together with a pose as input. They are trained to predict the correct pose based on the (potentially) noisy pose provided by the bottom-up methods (Figure 1).

Thus, BUCTD overcomes the information bottleneck and ambiguity introduced by standard detectors, while typically having similar or lower inference cost (Figure 1). We evaluate BUCTD on COCO [26], two crowded human benchmarks, CrowdPose [23] and OCHuman [43], and three multi-animal benchmarks, namely SchoolingFish, Tri-Mouse and Marmosets [22]. We achieve SOTA performance and strongly outperform both top-down and bottom-up models in occluded and crowded scenes.

## 2. Related Work

### 2.1. Multi-instance pose estimation & benchmarks

Top-down approaches detect the body parts of each individual by a single-instance pose estimation model [13, 8, 39] within the detected bounding box generated by the object detector [34, 33, 17, 12, 27]. Recently, transformer-based top-down methods such as TransPose [42], TokenPose [25], TFpose [28] and ViTPose [40] have increased the performance. Exemplar bottom-up approaches include OpenPose [6], Associative Embedding [31], ArtTrack [19], HigherHRNet [9], DEKR [16], DLCRNet [22], CID [38] and PETR [36]. Additionally, ensuring precision in pose estimation has led to the development of pose refinement methods. Works such as PoseRefiner [14] and PoseFix [30]

proposed models for refining the predicted pose (of a different model) and can substantially improve the accuracy.

Classic benchmarks for human pose estimation, such as COCO [26] and MPII [1], contain few occlusions [21], even though this is typical in many real-world scenarios. In recent years new benchmarks with more crowded scenes emerged, most notably CrowdPose [23] and OCHuman [43]. Interestingly, multi-animal pose estimation shares some of the challenges of human benchmarks, but also raises other problems [29], such as lack of “social” distancing amongst animals and highly similar appearances within a given species, such as mice. Therefore, to tackle these challenges we also focused on multi-animal benchmarks comprising mice, monkeys, and groups of fish with heavy overlap [22]. Our method, BUCTD, achieves SOTA on these benchmarks.

### 2.2. Crowded scene pose estimation

Many recent works [23, 43, 32, 21] have focused on occluded scenes in pose estimation. Khirodkar et al. [21] propose a hybrid top-down approach called MIPNet, which allows the model to predict multiple people within a given bounding box. MIPNet reached SOTA performance by providing an integer input to indicate which human with respect to the distance from the center should be predicted. In contrast, we hypothesize that providing a pose cue about which individual should be predicted is advantageous. The CID model [38] proposes an end-to-end architecture including a CNN backbone and a feature decoupling stage to distinguish between individuals. However, the decoupling is only based on the center prediction of individuals. CenterGroup [4] uses attention to link person centers to body parts, while PETR [36] deploys separate, transformer-based decoders for individuals and keypoints respectively. Ding et al. [11] recently proposed another attention-based model that appears to have strong performance on CrowdPose and OCHuman. However, the authors evaluate their model differently than the field, i.e., only based on ground-truth bounding boxes, and hence we do not compare it to other approaches.

### 2.3. Combining Top-Down and Bottom-Up Models

Hu and Ramanan [18] proposed a bidirectional architecture for hierarchical Rectified Gaussian models incorporating top-down feedback with a bottom-up architecture, while Tang et al. [37] introduced a hierarchical, compositional model, for which the inference process consists of both bottom-up and top-down stages across multiple semantic levels. Cai et al. [5] developed a graph-based method for 3D pose estimation by concatenating bottom-up features and top-down features together. Li et al. [24] proposed to use bottom-up methods to estimate the joints and leverage the bounding boxes from an object detector to group the

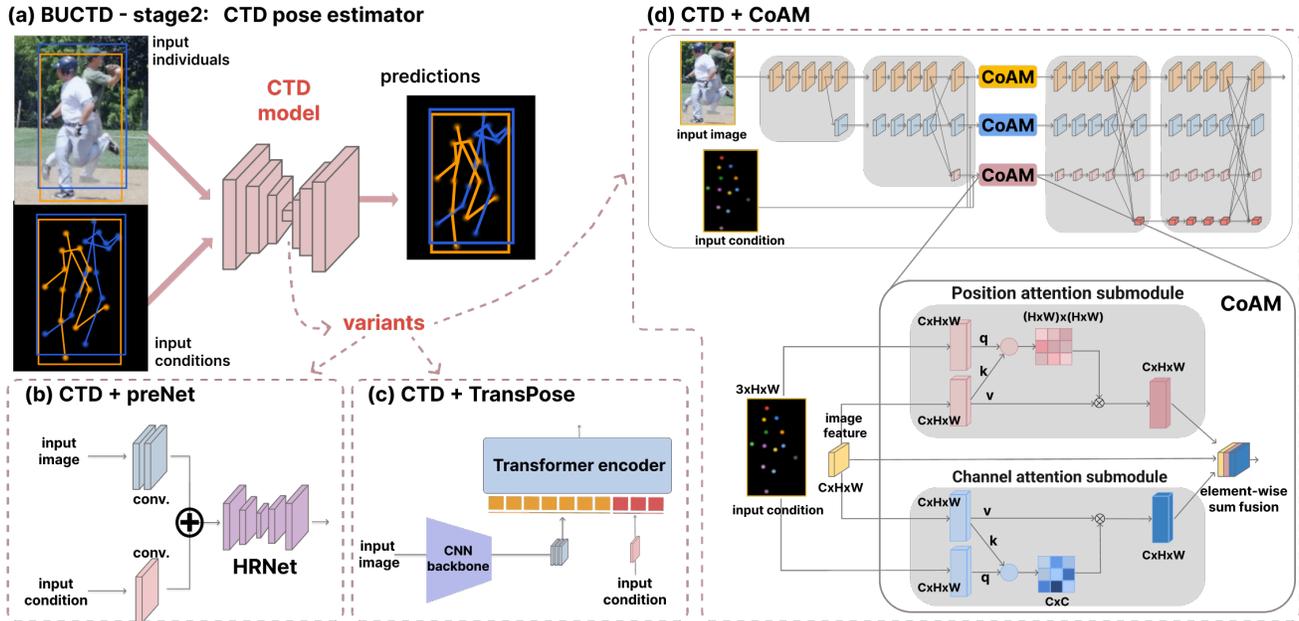


Figure 2. **Overview of the second stage of our BUCTD approach: conditional top-down (CTD) pose estimators.** (a) CTD receives a conditional pose and a cropped image, whose crop was estimated from the conditioned keypoints. (b) CTD with preNet. (c) CTD with TransPose. (d) CTD with Conditional Attention Module (CoAM). (**Top**: Our CTD model simply employs the same multi-resolution stages as the standard HRNet; we insert our CoAM module after stage 2 (one CoAM for every resolution branch). **Bottom**: CoAM. The output of stage 2 for one branch (image feature) is treated as keys and values for the two attention submodules, while we feed the color-coded condition heatmap as queries into the attention mechanisms. The output of the module is a combination of the spatial attention feature, the channel attention feature and the original image feature.

joints, while Cheng et al. [10] shows another, similar way to combine top-down and bottom-up approaches.

In comparison to previous works, instead of using an object detector, BUCTD leverages bottom-up models as detectors to provide a ‘pointer’ to guide the adapted TD model to pay attention to the correct target individual. For the first stage of BUCTD, we are building on the latest bottom-up methods, such as CID [38], and PETR [36]. For the second stage, we generalized top-down models to conditional top-down models (CTD). Instead of an index as in MIPNET [21], we provide a ‘pointer’ in the form of a pose predicted from a bottom-up model. Thus, our CTD model gets two inputs: a bounding box and a pose, similar in spirit to PoseFix [30] or PoseRefiner [14]. Akin to PoseFix and PoseRefiner, we also test a convolutional PreNet that provides the pose at the early stage of the top-down model. Moreover, we developed transformer and attention-based models for providing the conditional input. We find that these models achieve better performance and are efficient.

How is the conditional input provided to the second-stage model (during training)? We either sample it from known error distributions [35, 30], or based on the predictions of the BU models. As we will show, both approaches provide strong results.

## 3. Methods

### 3.1. The BUCTD framework

Our BUCTD model is a two-stage model trained to predict the pose from the cropped input image and the bottom-up input (Figure 2). The intuition is that the BUCTD model can use the image as well as the pose input to ‘know’ which individual to predict. This overcomes the inherent ambiguity for top-down approaches in crowded scenes. Importantly, our training scheme differs from classic top-down approaches, since we train on cropped images generated from bottom-up pose predictions, in contrast to using the ground-truth bounding boxes. This induces additional augmentation in the training.

**Stage1: Bottom-Up detector (BU).** Firstly, to detect individuals, we used bottom-up pose estimation models on the target training dataset and get the predictions. Classic top-down methods use a generic object detector to get the bounding boxes for individuals, however, bounding boxes create an information bottleneck between the detector and the pose estimation model. Furthermore, in real-world applications, training an object detector often has a higher computational cost compared to training bottom-up pose models (Suppl. Materials). Therefore, we propose to use

### (a) Qualitative results



### (b) Comparison results

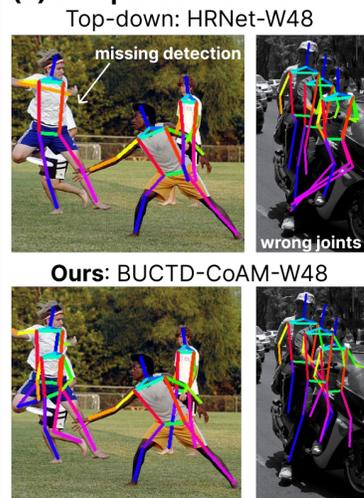


Figure 3. **CrowdPose and OCHuman, qualitative results.** (a) Qualitative results on CrowdPose (top row) and OCHuman (bottom row) with BUCTD-CoAM-W48. (b) Top: Predictions from a top-down approach (HRNet-W48) and bottom: predictions from our BUCTD model with CoAM-W48, both on CrowdPose images. Missing detections or wrongly predicted keypoints are noted.

a bottom-up model as the detector, which will provide both a predicted pose as guidance, and the corresponding bounding box. Of course, also recent single-stage models [36, 38] can (and will) be used.

**Stage2: Conditional Top-Down (CTD).** Secondly, we train CTD models with the conditional pose input and the corresponding bounding box. During training, we experiment with two different sampling strategies for the conditions. 1) **Empirical sampling:** We sample the conditions from the predictions of BU models, where we match those predictions to the ground truth pose by using Object Keypoint Similarity (OKS) on human benchmarks and bounding box IoU on animal benchmark. 2) **Generative sampling:** Instead of taking actual predictions, we also synthetically sample input poses based on estimated pose estimation errors [35]. This approach is similar to PoseFix [30].

## 3.2. CTD model architectures

The second stage of our BUCTD approach consists of the conditional top-down (CTD) pose estimation model, that provides a generic solution to tackle multi-instance pose estimation in crowded scenes. In this section, we describe the flexibility of the BUCTD approach by adopting different TD architectures (Figure 2a-d). In all cases, we convert the conditional input from the predicted keypoints into a 3-channel heatmap by using a Gaussian distribution with a standard deviation  $\sigma$ . To keep the semantic information, we provide each keypoint with a certain RGB value. We validate these design choices in the Suppl. Materials.

### 3.2.1 CTD with preNet

Given an input pair comprising an input image and input condition, we feed the pair in parallel to the preNet which contains two different convolutional layers, to extract features from both inputs (i.e., two  $7 \times 7$  conv. layers for input image and one  $7 \times 7$  conv. layer for the conditional input). Then, we fuse the image feature and the condition feature together and feed it into a HRNet [39]. Thus, CTD-preNet is a simple architecture leveraging the CTD approach (Figure 2b), which is similar to PoseFix’s design [30].

### 3.2.2 CTD with TransPose

We modified TransPose [42] to leverage its powerful transformer architecture. TransPose consists of a CNN backbone whose output features are transformed to  $d \times H \times W$  by a  $1 \times 1$  convolution. These  $d$ -dimensional image feature maps are then flattened into a sequence  $X \in \mathbb{R}^{N \times d}$ , with  $N = H \times W$ , which is then fed to a standard transformer encoder. For CTD+TransPose, we provide conditions as “side-information” by concatenating condition-specific tokens to this transformer input sequence (Figure 2c). Specifically, we transform the conditional heatmap first by a  $1 \times 1$  convolution to expand the number of channels to  $c$  and, after flattening, obtain a condition sequence  $Y \in \mathbb{R}^{N \times c}$ . Here, we arbitrarily chose  $c = 16$ . The final input sequence to the transformer encoder is the concatenation of  $X$  and  $Y$ , i.e.,  $X \oplus Y$ . The attention layers enable capturing long-range relationships between the conditional input and the predicted keypoints.

### 3.2.3 CTD with a Conditional Attention Module

To learn a better representation and leverage the information from the conditional input, we propose an architecture comprised of a Conditional Attention Module (CoAM) and HRNet [39], where we feed the input image and the corresponding pose condition into HRNet and the CoAM in parallel (Figure 2d). The CoAM can be inserted after any HRNet-stage and its output is fused with the features of the corresponding stage. Unless otherwise indicated the CoAM input was given to HRNet-stage 2 (see ablations). Therefore, the conditional input (1) provides a clue to which individual in the crop the CTD model should focus on, (2) improves the input pose obtained from the bottom-up model.

CoAM is inspired by [15] and contains spatial and channel attention sub-modules. It is designed to learn associations between features and “conditions” with an attention-like mechanism. Finally, it aggregates and fuses the features from both sub-modules by performing an element-wise sum. The resulting feature map is added back to the module’s input feature map, hence combining the features extracted from the HRNet with the long-range contextual information. The CoAM module treats the conditions as queries and the feature maps as keys and values for calculating the attention scores (Figure 2d). Details on the Position and Channel Attention Modules are provided in the Suppl. Materials.

### 3.3. Implementation & training details

**BUCTD with preNet.** We equipped HRNet-W32 or W48 [39] with a preNet to train BUCTD-preNet-W32 and W48, respectively, and report the efficiency on the animal benchmarks [22], CrowdPose [23] and COCO [26].

**BUCTD with TransPose.** We trained BUCTD-TP-H-A6 model (based on the TransPose-H-A6 architecture [42]) on CrowdPose to show that the conditional top-down approach can be successfully integrated into transformers. We ran this architecture on CrowdPose [23] and COCO [26].

**BUCTD with CoAM.** We trained HRNet-W32 and HRNet-W48 [39] with CoAM (BUCTD-CoAM-W32 and BUCTD-CoAM-W48) on all benchmarks [22, 23, 26, 43].

**Training Details:** To obtain the conditional inputs we trained bottom-up pose estimation models: specifically DL-CRNet [22] for animal pose, and HigherHRNet [9] for CrowdPose and OCHuman, and saved (pose) predictions from different model checkpoints (animals: up to 8-12 checkpoints, human: 15 checkpoints).

During training and inference, we added a fixed margin (25 pixels) in height and width to the predicted bounding box, for animal datasets. To keep the aspect ratio and avoid distortion of the animal’s body, we resized and padded the predicted bounding box to  $256 \times 256$ . For the human data, we used a margin of 5 pixels and extend each detection box

to a fixed aspect ratio ( $256 \times 192$  or  $384 \times 288$ ). We followed the same training scheme (batch size, learning rate, weight initialization, augmentation scheme, loss function) as in [39, 21] (see Suppl. Materials). As for the experiments using generative sampling strategy, we use the same error distribution as in PoseFix [35, 30] on human benchmarks, while for animal benchmarks we adapt it (Suppl. Materials).

## 4. Experiments

To evaluate BUCTD, we performed comprehensive experiments on several benchmarks. We tested our approach on the most important benchmarks for crowded scenes (CrowdPose [23] and Occluded Human (OCHuman) [43]), as well as on three multi-animal pose estimation benchmarks [22], and COCO [26]. We also carried out several ablations to test the design choices.

### 4.1. CrowdPose Benchmark

**Dataset:** The CrowdPose dataset [23] contains 12K labeled images in the *trainval* set with 43.4K labeled people (each with 14 keypoints), and 8K images in the *test* set with 29K labeled people. Following other studies [16, 36, 38], we used *trainval* for training, and *test* for evaluation. We report standard metrics AP, AP<sub>easy</sub>, AP<sub>med</sub> and AP<sub>hard</sub> as defined in [23]. We compared our method, that derives bounding boxes from a bottom-up model (see Methods), with baselines that used bounding boxes obtained by a Faster R-CNN detector [34].

**Results:** First, we trained the different CTD models on CrowdPose with BU predictions from HigherHRNet-W32 and empirical sampling. We found that all variants can boost the performance of HigherHRNet by up to 7 AP (Table 1). We evaluated the test-performance (without re-training) of CTD models when they were provided with inputs from recently described SOTA bottom-up or single-stage models (CID-W32, DEKR and PETR, respectively). We found that the CTD models could generalize to other bottom-up pose estimation model inputs (Table 1). Using the CoAM module provided the best results, and outperformed the CTD-preNet models, which is similar to PoseFix [30]. This highlights that CTD models are good pose refiners. Next we checked the performance of the full BUCTD pipeline.

We then compared BUCTD to bottom-up, single-stage, and top-down methods. Overall, BUCTD achieved SOTA performance on CrowdPose (Table 2, Figure 3). As we already showed, BUCTD can improve the performance over BU models, at the cost of additional computation (due to inference with CTD models). BUCTD also outperforms top-down and hybrid methods, while having comparable computational costs. Training and performing inference are comparable for object detectors and BU models, as BU

Method	AP	AP <sub>easy</sub>	AP <sub>med</sub>	AP <sub>hard</sub>
HigherHRNet-W32 [9]	65.7	73.2	66.1	57.9
CTD-preNet-W32†	69.5 (+3.8)	76.2	69.9	62.5 (+4.6)
CTD-TP-H-A6†	70.7 (+5.0)	77.9	71.1	63.0 (+5.1)
CTD-CoAM-W32†	71.4 (+5.7)	78.0	71.8	64.5 (+6.6)
CTD-CoAM-W48*	72.9 (+7.2)	79.2	73.4	66.1 (+8.2)
DEKR [16]	68.0	76.6	68.8	58.4
CTD-preNet-W32†	69.7 (+1.7)	77.7	70.6	60.5 (+2.1)
CTD-TP-H-A6†	71.0 (+3.0)	79.1	71.9	61.7 (+3.3)
CTD-CoAM-W32†	71.1 (+3.1)	78.8	71.9	61.8 (+3.4)
CTD-CoAM-W48*	72.0 (+4.0)	79.5	72.8	63.0 (+4.6)
CID-W32 [38]	71.3	77.4	72.1	63.9
CTD-preNet-W32†	72.8 (+1.5)	79.0	73.4	65.7 (+1.8)
CTD-TP-H-A6†	73.7 (+2.4)	80.1	74.5	66.2 (+2.3)
CTD-CoAM-W32†	74.2 (+2.9)	80.2	74.9	67.1 (+3.2)
CTD-CoAM-W48*	75.3 (+4.0)	81.1	75.9	68.4 (+4.5)
PETR [36]	72.0	78.0	72.5	65.4
CTD-preNet-W32†	74.6 (+2.6)	80.9	75.1	67.7 (+2.3)
CTD-TP-H-A6†	75.6 (+3.6)	82.2	76.1	68.6 (+3.2)
CTD-CoAM-W32†	75.9 (+3.9)	81.9	76.3	69.1 (+3.7)
CTD-CoAM-W48*	76.7 (+4.7)	82.6	77.2	70.4 (+5.0)

Table 1. CTD boosts CrowdPose results using conditional inputs from different bottom-up and single-stage models (on *test* set). † and \* denotes input resolution of 256x192 and 384x288 respectively. All CTD models are trained with HigherHRNet-W32 conditional input.

models often have similar or fewer parameters and GFLOPs (Suppl. Materials).

Strikingly, we improved upon MIPNet-W48 [21] by up to 8.5 AP. To achieve this, BUCTD-CoAM-W48 was trained with bottom-up predictions from HigherHRNet-W32 [9] (which alone performs relatively poorly) and generative sampling, as well as inputs from PETR [36]. We also compared to PoseFix [30], which did not evaluate on CrowdPose [23]. Thus, we implemented PoseFix by utilizing generative sampling and our BUCTD-preNet architecture, which is similar to the original PoseFix [30], but with a more powerful backbone. This PoseFix-HRNet-W48 achieves 76.8 AP, while BUCTD (with CoAM) achieves 78.5 AP. Interestingly, for BUCTD, we also achieved the best results with generative sampling. Collectively, this suggests that our hybrid approach combines the strengths of both bottom-up and top-down methods, and can outperform refinement methods.

Moreover, to gain additional insights, we computed the precision and recall for the BUCTD approach for different BU models on CrowdPose. We compare our model to the previous SOTA on CrowdPose: MIPNet [21]. Importantly, BUCTD has both higher recall and precision than MIPNet for all models (Suppl. Materials).

**Ablation Results:** Some BU models provide more predictions than detectors. To fairly compare, we also provided exactly the same number of detections from the bottom-up models as provided by the object detector. Despite this artificial constraint the performance of BUCTD was still significantly higher than the one of MIPNet [21], indicating that

Method	Input size	AP	AP <sub>easy</sub>	AP <sub>med</sub>	AP <sub>hard</sub>
Bottom-Up methods					
HRNet-W48 [39]	640	67.3	74.6	68.1	58.7
HigherHRNet-W48 [9]	640	67.6	75.8	68.1	58.9
CenterGroup [4]	640	69.4	76.6	70.0	61.5
BAPose [2]	512	72.2	79.9	73.4	61.3
Single-stage methods					
DEKR [16]	640	68.0	76.6	68.8	58.4
PETR [36]	800	72.0	78.0	72.5	65.4
CID-W32 [38]	512	71.3	77.4	72.1	63.9
CID-W48 [38]	640	72.3	78.7	73.0	64.8
Top-Down methods					
AlphaPose [13]	-	61.0	71.2	61.4	51.1
JC-SPPE [23]	320×256	66.0	75.5	66.3	57.4
HRNet-W48 [39]	384×288	69.3	77.7	70.6	57.8
Hybrid approaches					
MIPNet-W48 [21]	384×288	70.0	78.1	71.1	59.4
PoseFix (HRNet-preNet-W48)++ $\sigma$	384×288	76.8	82.3	77.4	70.2
BUCTD-CoAM-W48 (Ours)	384×288	72.9	79.2	73.4	66.1
BUCTD-CoAM-W48+ (Ours)	384×288	75.3	81.1	75.9	68.4
BUCTD-CoAM-W48++ (Ours)	384×288	76.7	82.6	77.2	70.4
<b>BUCTD-CoAM-W48++<math>\sigma</math> (Ours)</b>	384×288	<b>78.5</b>	<b>83.9</b>	<b>79.0</b>	<b>72.3</b>

Table 2. BUCTD improved performance on CrowdPose *test* set. For empirical sampling, BUCTD models are trained with HigherHRNet-W32 conditions. + denotes cond. input from CID-W32, ++ denotes cond. input from PETR, and  $\sigma$  denotes generative sampling.

the performance gains are not simply coming from a higher number of provided detections (Suppl. Materials).

To validate our design choices we performed ablation studies on the CTD-CoAM-W32 model. We experimented with the position of CoAM and found that conditional inputs are best provided at earlier stages in the network (Table 3). We further wanted to validate the impact of different conditional inputs during training. In comparison to the 15 checkpoints we obtained from training the bottom-up model (HigherHRNet-W32), we trained CTD models with conditions from different numbers and types of conditions. Generally speaking, CTD models can better learn to predict poses when they are trained with diverse conditional poses for empirical sampling. Accordingly, generative sampling can further improve the performance (Table 4).

Method	stage	AP	AP <sub>easy</sub>	AP <sub>med</sub>	AP <sub>hard</sub>
BUCTD-CoAM-W32	stage 1	<b>71.2</b>	<b>77.8</b>	<b>71.6</b>	64.2
BUCTD-CoAM-W32	stage 2	<b>71.2</b>	77.7	<b>71.6</b>	<b>64.3</b>
BUCTD-CoAM-W32	stage 3	70.5	77.4	70.9	63.3
BUCTD-CoAM-W32	stage 4	68.3	76.8	69.2	58.7

Table 3. Ablating the position of CoAM on CrowdPose *test*. Feeding the conditions through CoAM in earlier stages of the HRNet architecture increased performance.

## 4.2. OCHuman Benchmark

**Dataset:** The OCHuman dataset [43] is the most challenging dataset for crowded multi-person pose estimation with an average of 0.67 MaxIoU (Intersection over Union

GT	#checkpts	#checkpts	#checkpts	#checkpts	AP	AP <sub>easy</sub>	AP <sub>med</sub>	AP <sub>hard</sub>	AP	AP <sub>easy</sub>	AP <sub>med</sub>	AP <sub>hard</sub>
	HrHRNet	DEKR	CID	PETR	Tested on HrHRNet conditions				Tested on PETR conditions			
-	1 (best)	-	-	-	68.8	75.8	69.3	61.5	73.9	80.4	74.4	67.1
-	1 (test-gap)	-	-	-	69.8	76.6	70.3	62.8	74.7	80.9	75.1	67.9
-	1	1	1	1	68.9	76.0	69.4	61.5	74.4	80.7	74.8	67.6
-	15	-	-	-	71.4	78.0	71.8	64.5	75.9	81.9	76.3	69.1
1	15	1	1	1	70.9	77.7	71.5	63.9	75.5	81.7	75.9	68.9
+gen.	-	-	-	-	72.3	78.8	72.8	65.1	76.9	83.0	77.4	70.4

Table 4. **Additional ablation studies for the number and type of different checkpoints during training on CrowdPose.** For empirical sampling, higher diversity (predictions sampled from different checkpoints of the bottom-up model) leads to better performance. We also found that generative sampling works well. “Best” denotes the best performing checkpoint, while “test-gap” denotes the one for which training performance is closest to the final testing performance. All models are based on a HRNet-W32, are trained on input resolutions of 256x192 and tested with flipping.

between bounding boxes) for each person. It contains 4,731 images with 8,110 persons in total. For a fair comparison, we report the results in the same way as illustrated in [43, 38], namely we train our models on the COCO *train* set and evaluate on OCHuman *test* set.

**Results:** Previous SOTA results on OCHuman were reported by the single-stage model CID [38]. We reach new SOTA performance, with gains up to 3.5 AP. Notably even BUCTD with a smaller HRNet outperforms MIPNet. Naturally, the BUCTD pipeline improves over plain bottom-up methods, which historically do well in crowded scenes, and over the recently introduced strong single-stage method CID [38]. Qualitative results can be seen in Figure 3.

Method	AP val	AP test
HGG [20]	35.6	34.8
HigherHRNet-W32 [9]	40.0	39.4
LOGO-CAP-W48 [41]	41.2	40.4
DEKR [16]	37.9	36.5
CID-W32 [38]	45.7	44.6
CID-W48 [38]	46.1	45.0
AlphaPose+ [32]	-	27.5
HRNet-W48* [39]	37.8	37.2
MIPNet-W48* [21]	42.0	42.5
BUCTD-CoAM-W32 (HrHRNet-W32)†	44.1	43.5
BUCTD-CoAM-W32 (CID-W32)†	47.3	46.3
BUCTD-CoAM-W48 (CID-W32)σ†	48.3	47.4
BUCTD-CoAM-W48 (CID-W32)σ† 2x	48.8	48.3
BUCTD-CoAM-W48 (CID-W32)σ† 3x	<b>49.0</b>	<b>48.5</b>

Table 5. **BUCTD improved performance on OCHuman.** Comparison with state-of-the-art methods on the OCHuman val and test set after training on COCO train. † and \* denotes input resolution of 256x192 and 384x288, respectively. Model in brackets denotes where conditions are coming from during inference. 2x and 3x marks iterative refinement by feeding back model predictions as new conditions. For empirical sampling, BUCTD models are trained with HigherHRNet-W32 conditions.  $\sigma$  denotes generative sampling.

### 4.3. COCO Benchmark

**Dataset:** The COCO [26] dataset contains 57K images with 150K persons in the *train* set, 5K images with 6.3K persons in the *val* set and 20K images in the test-dev set. We used *train* for training and *val* for validation. We compared our method with several bottom-up models and top-down methods, but note that it has few overlapping people [21] compared to CrowdPose [23] and OCHuman [43].

**Results:** BUCTD can refine the performance of single-stage, BU and TD (HRNet-W48) methods. This suggests that our method, designed for the challenges of crowdedness, can also fare well even with few overlapping people (Table 6).

Method	AP	AP <sub>M</sub>	AP <sub>L</sub>
DEKR [16]	71.0	66.7	78.5
CID-W32 [38]	69.8	64.0	78.9
PETR [36]	73.1	67.2	81.7
HRNet-W48 [39]	76.3	72.3	83.4
MIPNet-W48 [21]	76.3	72.3	83.4
PoseFix [30] (best original)σ	73.6	70.3	79.8
PoseFix (our HRNet-preNet-W48)σ	77.3	73.5	84.4
BUCTD-CoAM-W48 (DEKR)σ	74.8	71.1	81.1
BUCTD-CoAM-W48 (CID-W32)σ	74.8	71.1	81.1
BUCTD-CoAM-W48 (PETR)σ	77.1	73.3	83.4
<b>BUCTD-preNet-W48 (PETR)σ</b>	<b>77.8</b>	<b>74.2</b>	<b>83.7</b>
BUCTD-TP-H-A6 (PETR)σ	76.0	72.2	82.3
BUCTD-CoAM-W48 (HRNet-W48)σ	76.5	72.7	83.2

Table 6. **Results on COCO val set.** BUCTD can be used to refine poses from other models also on COCO. Models trained with generative sampling are denoted by  $\sigma$ .

### 4.4. Multi-Animal Benchmarks

**Datasets:** To further assess the performance of BUCTD, we evaluated it on multi-animal benchmarks by Lauer et al. called SchoolingFish, Marmosets, and Tri-Mouse [22].

These datasets do not contain ground-truth bounding boxes. Thus, we trained the TD models in the same way as

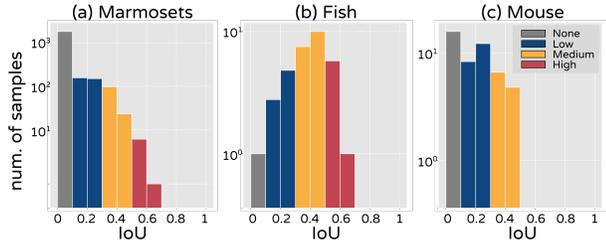


Figure 4. **Crowdedness levels for the animal datasets**, as divided into low, medium, and hard. The SchoolingFish dataset (Fish) is overall more crowded, while the Tri-Mouse dataset (Mouse) is least crowded (IoU: Intersection Over Union).

CTD models, i.e. bounding boxes of train and test samples were computed from the predictions of DLCRNet. To compare the classic TD pipeline with an object detector, we also ran the TD models in a traditional way: training with ground truth bounding boxes (derived from GT poses), and testing the model with the bounding boxes from the object detectors (Faster R-CNN [34] and YOLOv3 [33], as on CrowdPose). We evaluated BUCTD-W32, -W48 and compared them to TD and BU methods.

**Evaluation for different crowdedness levels.** These datasets vary in crowdedness (Figure 4). We calculated the Intersection Over Union (IoU) of the overlapped bounding boxes in the images and use the maximum IoU for each image (i.e., maxIoU) to indicate the crowdedness level (we show the analysis results in Suppl. Materials). We then further split the test set into different crowdedness levels for the test set: low ( $S_L$ ), medium ( $S_M$ ), and high ( $S_H$ ); Figure 4) to compute the metrics  $AP_L$ ,  $AP_M$ ,  $AP_H$ , respectively, to interrogate the model performance for different crowdedness levels test set (Table 7).

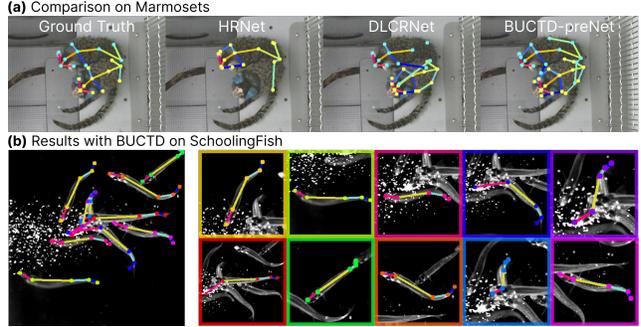


Figure 5. **Qualitative results on animal datasets.** (a) Each image with two marmosets shows the GT pose, HRNet-W48 predictions, DLCRNet predictions and BUCTD-preNet-W48 predictions (left to right). Due to crowdedness and occlusions, HRNet-W48 often missed the whole individual. DLCRNet may assign the body part to the wrong individual. BUCTD approach shows better performance in highly crowded conditions. (b) Results with BUCTD-preNet-W48 on one zoomed in fish image.

**Results:** We trained BUCTD models using the predictions from a bottom-up method based on DeepLabCut’s DLCRNet [22]. BUCTD outperforms the baseline methods on mice and marmosets and is competitive for fish. Importantly, BUCTD outperforms all other methods for crowded frames (see  $AP_H$ , Table 7, and Figure 5).

Additionally, we trained the CTD models using the generative sampling scheme. During testing, we use the same pose inputs as for BUCTD models. These models also improve the performance on the three animal benchmarks (Table 7). However, compared to sampling the conditions from BU predictions (empirical sampling), this sampling strategy only performs well for marmosets and mice. We speculate

Method (detector)	Marmosets				SchoolingFish				Tri-Mouse			
	AP	$AP_L$	$AP_M$	$AP_H$	AP	$AP_L$	$AP_M$	$AP_H$	AP	$AP_L$	$AP_M$	$AP_H$
<b>Bottom-up methods</b>												
Resnet [22]-AE [31]	45.0	-	-	-	40.0	-	-	-	70.3	-	-	-
HRNet [22]-AE [31]	65.1	-	-	-	45.7	-	-	-	83.9	-	-	-
DLCRNet [22]	80.1	75.3	71.9	67.5	74.1	68.7	77.6	72.8	95.8	94.9	94.7	-
CID-W32 [38]	92.5	90.5	91.8	82.9	81.0	72.7	84.0	79.9	86.8	84.1	85.6	-
DEKR [16]	61.4	64.8	64.8	57.7	77.6	82.3	80.3	72.1	97.2	97.2	97.9	-
PETR [36]	93.2	89.7	89.6	70.4	79.3	72.3	71.8	80.8	82.3	84.1	78.7	-
<b>Top-down methods</b>												
HRNet-W48 [39] (YOLOv3 [33])	91.0	90.1	87.7	44.1	82.9	80.6	85.2	79.2	91.7	91.2	89.5	-
HRNet-W48 [39] (Faster R-CNN [34])	91.6	90.2	85.0	42.2	<b>89.1</b>	82.8	<b>92.6</b>	86.1	96.0	97.0	90.3	-
<b>Hybrid approaches</b>												
BUCTD-preNet-W48 (DLCRNet)	90.4	87.0	86.1	85.7	88.7	<b>85.8</b>	90.5	88.9	98.5	97.9	98.3	-
BUCTD-preNet-W48 (CID-W32)	93.3	<b>91.9</b>	<b>93.4</b>	<b>89.9</b>	88.0	79.3	90.4	<b>90.5</b>	87.7	85.6	87.3	-
BUCTD-CoAM-W48 (PETR)	<b>93.7</b>	91.3	90.6	73.8	78.8	73.8	71.9	81.0	82.7	84.1	79.5	-
BUCTD-CoAM-W48 (DLCRNet) $\sigma$	91.6	86.3	88.9	89.4	81.9	71.0	81.0	78.3	<b>99.1</b>	<b>99.1</b>	<b>99.2</b>	-

Table 7. **BUCTD performance on Animal Pose Datasets.** BUCTD model largely outperforms top-down methods in crowded scenes. Model in brackets denotes where conditions are coming from during inference. Here,  $\sigma$  denotes the generative sampling training scheme, while others used empirical sampling is based on DLCRNet predictions.

Method	Marmosets		SchoolingFish		Tri-Mouse	
	AP	$AP_H$	AP	$AP_H$	AP	$AP_M$
HRNet-W48 (YOLOv3)	91.0	44.1	82.9	79.2	91.7	89.5
BUCTD-CoAM-W32	91.3	45.5	<b>87.8</b>	<b>85.0</b>	94.9	91.5
BUCTD-CoAM-W48	91.3	48.0	86.3	82.1	92.4	90.5
BUCTD-preNet-W48	91.8	50.4	85.2	79.5	91.8	89.4
BUCTD-CoAM-W48 $\sigma$	<b>93.1</b>	<b>52.4</b>	79.5	74.0	<b>96.9</b>	<b>93.8</b>
HRNet-W48 (FasterRCNN)	91.6	42.2	89.1	86.1	96.0	90.3
BUCTD-CoAM-W32	91.8	42.7	90.8	88.0	96.4	90.6
BUCTD-CoAM-W48	91.6	44.8	<b>90.9</b>	<b>88.7</b>	96.1	89.9
BUCTD-preNet-W48	91.8	44.8	89.3	87.4	96.3	91.9
BUCTD-CoAM-W48 $\sigma$	<b>92.8</b>	<b>49.8</b>	85.0	81.0	<b>97.3</b>	<b>92.5</b>
DLCRNet	80.1	67.5	74.1	72.8	95.8	94.7
BUCTD-CoAM-W32	89.5	84.3	86.9	86.5	98.4	97.6
BUCTD-CoAM-W48	89.5	84.5	88.2	86.6	98.5	98.3
BUCTD-preNet-W48	90.4	85.7	<b>88.7</b>	<b>88.9</b>	98.5	98.3
BUCTD-CoAM-W48 $\sigma$	<b>91.6</b>	<b>89.4</b>	81.9	78.3	<b>99.1</b>	<b>99.2</b>
CID-W32	92.5	82.9	81.0	79.9	86.8	85.6
BUCTD-CoAM-W32	93.1	84.0	86.3	85.4	90.9	88.0
BUCTD-CoAM-W48	92.9	86.3	87.2	83.8	90.2	88.1
BUCTD-preNet-W48	<b>93.3</b>	89.9	<b>88.0</b>	<b>90.5</b>	87.7	87.3
BUCTD-CoAM-W48 $\sigma$	91.8	<b>90.9</b>	83.2	81.1	<b>94.1</b>	<b>92.0</b>

Table 8. **Generalization results on animal benchmarks.** All CTD models boosted predictions from different models provided as conditional inputs. BUCTD models trained on DLCRNet predictions.  $\sigma$  denotes generative sampling.

that this is due to the different body plans of mammals and fish.

**Ablation Results:** One key question is if BUCTD is better because it is based on BU as a detector vs. a standard detector or due to its pose-refinement ability. We find that for crowded scenes, BU detections are key. We fed the pose predictions from top-down methods with standard detectors (YOLOv3, Faster R-CNN), to different CTD models and found that CTD can further improve the performance (Table 8). This validates that they can refine poses (akin to Table 1 for CrowdPose). Crucially, when considering the most crowded test data (i.e.  $AP_H$  for fish and marmosets and  $AP_M$  for mice) then the results were substantially worse than for the full BUCTD pipeline. For example, on the difficult marmoset frames detector with CTD only reached 52.4 mAP vs. 89.4 with BUCTD-CoAM-W48 (DLCRNet). The same is true for other animals and models (Table 8, and Figure 1).

Next, to validate our design choices on the bounding boxes during training, we performed ablation studies for the BUCTD-CoAM-W48 model using the animal benchmarks. The performance of top-down methods is influenced by the quality of the object detection outputs, i.e., the bounding boxes. Different from typical top-down methods, which train the model based on ground truth bounding boxes, and test on detected bounding boxes, we trained the BUCTD model using bounding boxes computed from bottom-up predictions. Validating this design, models trained on bottom-up-computed boxes perform the best on bottom-up-computed boxes during testing (Table 9).

Method	Training box		Marmosets	SchoolingFish	Tri-Mouse
	GT	BU	AP	AP	AP
HRNet-W48	✓	-	85.2	73.0	97.5
HRNet-W48	-	✓	<b>87.8</b>	<b>76.0</b>	<b>98.0</b>
BUCTD-CoAM-W48	✓	-	85.8	78.9	98.3
BUCTD-CoAM-W48	-	✓	<b>90.4</b>	<b>88.7</b>	<b>98.5</b>

Table 9. **Effect of bounding boxes on training.** Using bottom-up-computed (BU) boxes during training makes models more robust to crops seen during inference. This effect is particularly large for the BUCTD pipeline, validating the choice of training on bounding boxes that are computed with BU predictions (empirical sampling) or GT + error (generative sampling).

## 5. Discussion

Humans and other animals often interact closely making monocular pose estimation a challenging task. The field is increasingly creating benchmarks that encompass these challenges [23, 43, 22], and new solutions for accurately estimating poses in crowded scenes are actively being developed. Here, we presented a new hybrid approach to improve multi-instance pose estimation especially in crowded scenarios, which we call BUCTD. We compared BUCTD against classic bottom-up, top-down, refinement (PoseFix), and the recent hybrid top-down method (MIPNet), the latter showed excellent performance on crowded frames on challenging human benchmarks. Our BUCTD method achieves state-of-the-art performance on both CrowdPose and OCHuman and new animal pose benchmarks. We believe our proposal of using conditional input provided by a bottom-up pose estimator effectively helps in crowded scenes.

Of course, there still remains a gap on all the benchmarks considered. To further enhance performance, future work could additionally leverage hybrid sampling, or model-confidence across keypoints to condition the input predictions. We could also extend BUCTD to tracking of individuals and re-identification across frames. It could potentially also be adapted into video-based architectures, multi-modal learning settings, and with new optimization algorithms [7, 3].

In summary, BUCTD is a simple yet effective method that is model-backbone-agnostic. The usage of conditional input provided by a bottom-up pose estimator effectively helps solve heavily occluded and crowded scenes. Thus, we hope it can be a useful addition and broadly applied to newer architectures in computer vision as they arise.

## Acknowledgements

We are grateful to EPFL and EPFL’s School of Life Sciences PTECH fund for providing funding (AM), and the Vallee Foundation (MWM). We thank Steffen Schneider, Shaokai Ye, Haozhe Qi, and other members of the M.W.Mathis Lab and Mathis Group for feedback.

## References

- [1] 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. [2](#)
- [2] Bruno Artacho and Andreas Savakis. Bapose: Bottom-up pose estimation with disentangled waterfall representations. *arXiv preprint arXiv:2112.10716*, 2021. [6](#)
- [3] Jeremy Bernstein, Chris Mingard, Kevin Huang, Navid Azizan, and Yisong Yue. Automatic gradient descent: Deep learning without hyperparameters, 2023. [9](#)
- [4] Guillem Brasó, Nikita Kister, and Laura Leal-Taixé. The center of attention: Center-keypoint grouping via attention for multi-person pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11853–11863, 2021. [2](#), [6](#)
- [5] Yujun Cai, Lihao Ge, Jun Liu, Jianfei Cai, Tat-Jen Cham, Junsong Yuan, and Nadia Magnenat Thalmann. Exploiting spatial-temporal relationships for 3d pose estimation via graph convolutional networks. In *ICCV*, pages 2272–2281, 2019. [2](#)
- [6] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields, 2019. [2](#)
- [7] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms. *ArXiv*, abs/2302.06675, 2023. [9](#)
- [8] Zexin Chen, Ruihan Zhang, Yu Eva Zhang, Haowen Zhou, Hao-Shu Fang, Rachel R Rock, Aneesh Bal, Nancy Padilla-Coreano, Laurel Keyes, Kay M Tye, et al. Alphatracker: a multi-animal tracking and behavioral analysis tool. *biorxiv*, 2020. [2](#)
- [9] 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. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5386–5395, 2020. [2](#), [5](#), [6](#), [7](#)
- [10] Yu Cheng, Bo Wang, and Robby T Tan. Dual networks based 3d multi-person pose estimation from monocular video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1636–1651, 2022. [3](#)
- [11] Yiwei Ding, Wenjin Deng, Yinglin Zheng, Pengfei Liu, Meihong Wang, Xuan Cheng, Jianmin Bao, Dong Chen, and Ming Zeng. I<sup>2</sup>r-net: Intra- and inter-human relation network for multi-person pose estimation. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 855–862. ijcai.org, 2022. [2](#)
- [12] Glenn Jocher et. al. ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support. *Zenodo*, Oct. 2021. [1](#), [2](#)
- [13] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation, 2018. [2](#), [6](#)
- [14] Mihai Fieraru, Anna Khoreva, Leonid Pishchulin, and Bernt Schiele. Learning to refine human pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 205–214, 2018. [2](#), [3](#)
- [15] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3146–3154, 2019. [5](#)
- [16] Zigang Geng, Ke Sun, Bin Xiao, Zhaoxiang Zhang, and Jingdong Wang. Bottom-up human pose estimation via disentangled keypoint regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. [2](#), [5](#), [6](#), [7](#), [8](#)
- [17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. [2](#)
- [18] Peiyun Hu and Deva Ramanan. Bottom-up and top-down reasoning with hierarchical rectified gaussians. In *CVPR*, pages 5600–5609, 2016. [2](#)
- [19] Eldar Insafutdinov, Mykhaylo Andriluka, Leonid Pishchulin, Siyu Tang, Evgeny Levinkov, Bjoern Andres, and Bernt Schiele. Articulated multi-person tracking in the wild. *CoRR*, abs/1612.01465, 2016. [2](#)
- [20] 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 *European Conference on Computer Vision*, pages 718–734. Springer, 2020. [7](#)
- [21] Rawal Khirodkar, Visesh Chari, Amit Agrawal, and Ambrish Tyagi. Multi-instance pose networks: Rethinking top-down pose estimation. In *ICCV*, 2021. [2](#), [3](#), [5](#), [6](#), [7](#)
- [22] Jessy Lauer, Mu Zhou, Shaokai Ye, William Menegas, Stefan Schneider, Tanmay Nath, Mohammed Mostafizur Rahman, Valentina Di Santo, Daniel Soberanes, Guoping Feng, et al. Multi-animal pose estimation, identification and tracking with deeplabcut. *Nature Methods*, 19(4):496–504, 2022. [2](#), [5](#), [7](#), [8](#), [9](#)
- [23] 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*, pages 10863–10872, 2019. [2](#), [5](#), [6](#), [7](#), [9](#)
- [24] Miaopeng Li, Zimeng Zhou, and Xinguo Liu. Multi-person pose estimation using bounding box constraint and lstm. *IEEE Transactions on Multimedia*, 21(10):2653–2663, 2019. [2](#)
- [25] Yanjie Li, Shoukui Zhang, Zhicheng Wang, Sen Yang, Wankou Yang, Shu-Tao Xia, and Erjin Zhou. Tokenpose: Learning keypoint tokens for human pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11313–11322, 2021. [2](#)
- [26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. [2](#), [5](#), [7](#)

- [27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. [1](#), [2](#)
- [28] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, and Zhibin Wang. Tfpote: Direct human pose estimation with transformers. *arXiv preprint arXiv:2103.15320*, 2021. [2](#)
- [29] Mackenzie Weygandt Mathis and Alexander Mathis. Deep learning tools for the measurement of animal behavior in neuroscience. *Current opinion in neurobiology*, 60:1–11, 2020. [2](#)
- [30] 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](#), [3](#), [4](#), [5](#), [6](#), [7](#)
- [31] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. *Advances in neural information processing systems*, 30, 2017. [2](#), [8](#)
- [32] 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](#), [7](#)
- [33] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. [1](#), [2](#), [8](#)
- [34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. [1](#), [2](#), [5](#), [8](#)
- [35] Matteo Ruggero Ronchi and Pietro Perona. Benchmarking and error diagnosis in multi-instance pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 369–378, 2017. [3](#), [4](#), [5](#)
- [36] Dahu Shi, Xing Wei, Liangqi Li, Ye Ren, and Wenming Tan. End-to-end multi-person pose estimation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11069–11078, 2022. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [37] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *ECCV*, pages 190–206, 2018. [2](#)
- [38] Dongkai Wang and Shiliang Zhang. Contextual instance decoupling for robust multi-person pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11060–11068, 2022. [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [39] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition, 2020. [2](#), [4](#), [5](#), [6](#), [7](#), [8](#)
- [40] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. ViTPose: Simple vision transformer baselines for human pose estimation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. [2](#)
- [41] Nan Xue, Tianfu Wu, Gui-Song Xia, and Liangpei Zhang. Learning local-global contextual adaptation for multi-person pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13065–13074, 2022. [7](#)
- [42] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11802–11812, 2021. [2](#), [4](#), [5](#)
- [43] Song-Hai Zhang, Ruilong Li, Xin Dong, Paul Rosin, Zixi Cai, Xi Han, Dingcheng Yang, Haozhi Huang, and Shi-Min Hu. Pose2seg: Detection free human instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 889–898, 2019. [2](#), [5](#), [6](#), [7](#), [9](#)