Supplementary Document: Explicit Occlusion Reasoning for Multi-person 3D Human Pose Estimation

Qihao Liu[1], Yi Zhang[1], Song Bai[2], and Alan Yuille[1]

[1] Johns Hopkins University
[2] ByteDance

Abstract. In this supplementary document, we provide details and extended evaluations omitted from the main paper for brevity. Sec. 1 gives additional method details, Sec. 2 provides implementation details, Sec. 3 and 4 contain extended experimental evaluations, and Sec. 5 provides more detailed discussions on limitations and failure cases.

1 Additional Method Details

1.1 Root Depth Reasoning

As mentioned in the main paper, the root depth is inferred by the geometry of human joints instead of a deep neural network. We divide human joints into two classes: torso joints (including head, neck, pelvis, shoulders, and hips) and limb joints. The depth of pelvis can be inferred by the depth of visible torso joints.

To do so, we define two different kinds of symmetry pairs based on their expected errors when estimating root depth (Fig. 1). For example, the pelvis is defined to be the center of two hips in the MPI15 joint definition, which means given the positions of two hips, we can estimate the pelvis depth without error. Therefore, they are defined as the first-class symmetry pair. During inference, the

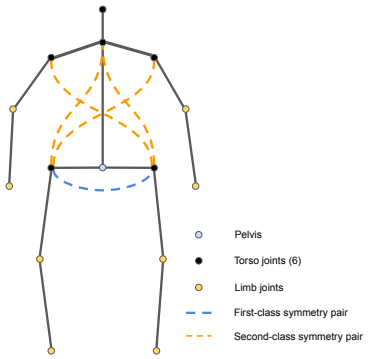
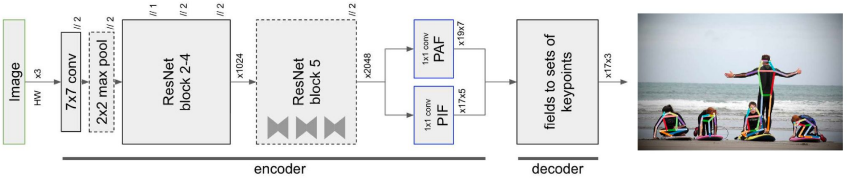
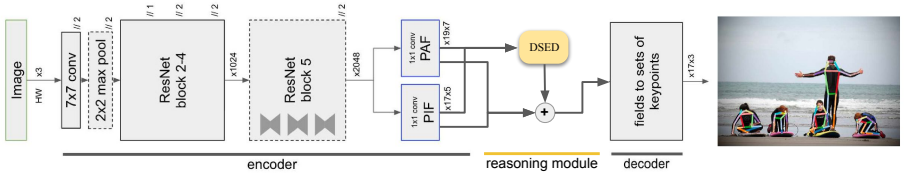


Fig. 1: The root depth can be inferred by the torso joints.



(a) Model architecture of original PifPaf



(b) Model architecture of PifPaf with our reasoning module

Fig. 2: Apply our occluded keypoints reasoning module to PifPaf.

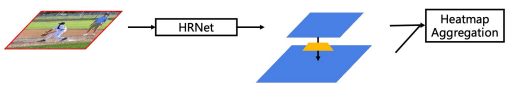
detection module outputs the 3D positions of all visible torso joints (including pelvis) and their confidence maps. Based on the confidence maps, we can first select all joint predictions with high confidence. If the pelvis is predicted with high confidence, then we directly use its estimate as root depth, otherwise, we select the visible symmetry pairs following the pre-defined order. After that, the root depth is computed based on the detected symmetry pairs.

1.2 DSED-based Reasoning Module on PifPaf

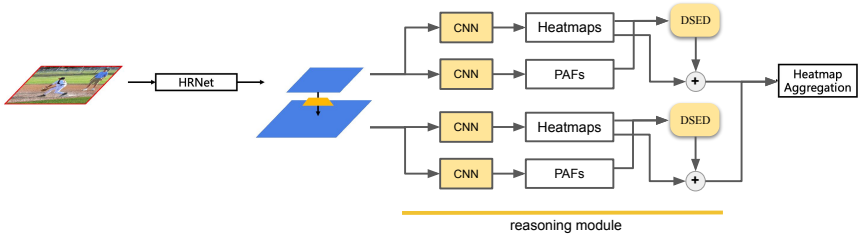
PifPaf [8] is a bottom-up method for multi-person 2D HPE. It uses a Part Intensity Field (PIF) to localize body parts and a Part Association Field (PAF) to associate body parts with each other to form full human poses. At every output location (i, j), a PIF predicts a confidence c and a vector (x, y) with spread b and scale σ. Combining all information of a PIF, we can get a more accurate heatmap prediction (See [8]). PAFs are slightly different from the PAFs in HUPOR, but they also provide skeleton information, thus we can still directly use them. Therefore, the encoder of PifPaf provides all the intermediate results we need for the reasoning module. We can directly insert our DSED-based reasoning module between the encoder and the decoder (Fig. 2). During training, we first train the network without the reasoning module following the settings in [8], but only provide supervision for the visible keypoints, then we freeze the rest part of the network and train the reasoning module with only occluded keypoints. When training the reasoning module, we set $\lambda_k^{occ} = \lambda_p^{occ} = \lambda_k^{all} = \lambda_p^{all} = 0.5$ and we linearly anneal $\omega_{extract}$ from 0.0 to its full value of 2.0 over the first half of the training process. Other settings of training the reasoning module follow HUPOR. Note that we only use a one-stacked DSED model here.

1.3 DSED-based Reasoning Module on HigherHRNet

Different from PifPaf, HigherHRNet [2] uses associative embedding for keypoint grouping. Therefore, to apply our reasoning module to it, we need first add a



(a) Model architecture of original HigherHRNet



(b) Model architecture of HigherHRNet with our reasoning module

Fig. 3: Apply our occluded keypoints reasoning module to HigherHRNet.

new branch to the network to output PAFs-like intermediate results, then we can apply our reasoning module (Fig. 3). During training, we first train the network without the reasoning module following [2], with a weight of 0.5 added to the loss of PAFs. Then we freeze the rest part of the network and train the reasoning module with only occluded keypoints. We set $\lambda_k^{occ} = \lambda_p^{occ} = \lambda_k^{all} = \lambda_p^{all} = 0.1$ and we linearly anneal $\omega_{extract}$ from 0.0 to its full value of 5.0 over the first half of the training process. Other settings of training the reasoning module follow HUPOR. Note that we only use a one-stacked DSED model here.

2 Implementation Details

2.1 HUPOR

The visible keypoint detection module is a three-stacked hourglass model [11] and the occluded keypoint reasoning module is a two-stacked DSED model. Both of them use Adam [6] with a learning rate of 2×10^{-4} and weight decay of 8×10^{-6} as the optimizer. The detection module is trained for $10+10$ epochs and in the first 10 epochs, it is trained alone without the reasoning module. The reasoning module is trained together with the detection module in the second 10 epochs. Similar to [15,14,21], we use a batch size of 32, and 50% of the data in each mini-batch is from COCO2017 [12]. All the images are resized to 832×512. The weights of 3D losses are set to 0 when data from COCO2017 is fed.

During training, $\lambda_k^{vis} = \lambda_p^{vis} = \lambda_k^{occ} = \lambda_p^{occ} = \lambda_k^{all} = \lambda_p^{all} = 0.1$, $\lambda_r^{vis} = 10$, and a scale factor of 50 is applied to the z dimension of the PAFs. We linearly anneal $\omega_{extract}$ from 0.0 to its full value of 1.0 over the first quarter of the training process.

Different from [21] that uses fully-connected layers as RefineNet, the RefineNet we use consists of Graph Convolution Networks (GCN) with 4 hidden layers and hidden sizes of 128. We compare their difference in Sec. 3.4. Training

Table 1: **Performance and efficiency on MuPoTS-3D.** We significantly surpass SOTA top-down [16] and bottom-up [21] methods in almost all metrics while remains high efficiency. 2Hg represents a 2-stacked hourglass model. When comparing model efficiency, we only consider the process of keypoint detection and reasoning. (*i.e.*, the main difference of HUPOR and SMAP)

	Matched people			All people		Efficiency (3-people)		Efficiency (20-people)	
	PCK_{abs}	PCK_{rel}	PCK_{occ}	PCK_{abs}	PCK_{rel}	Time(ms)	Memory(M)	Time(ms)	Memory(M)
Moon *et al.* [16]	31.8	82.5	66.8	31.5	**81.8**	147.7	2517	250.7	3441
SMAP (4Hg) [21]	38.6	80.9	73.1	35.2	73.7	87.1	1537	95.4	1537
Ours (2Hg+2DSED)	**39.0**	**85.9**	**74.6**	**35.9**	78.3	99.8	1661	104.7	1661

is performed using batches of 1024 poses for 200 epochs with Adam and setting $lr = 1 \times 10^{-2}$. Other settings follow [21].

2.2 SSF (and Details of ShapeInit, Skeleton2Pose, ShapeOpt)

For a fair comparison, we follow [10] and use ResNet-34 [3] as the network backbone, followed by fully-connected layers with 2 hidden layers and hidden sizes of (1024, 512), and a final layer to match with the output dimensions, to build the ShapeInit model. The Skeleton2Pose model is adopted from [10]. The ShapeOpt model consists of 4-layer CNN to handle the input mask, followed by fully-connected layers with 2 hidden layers and hidden sizes of (1024, 512). Then the output of the final layer is concatenated with the parameters of the reconstructed mesh. After that, fully-connected layers with 2 hidden layers and hidden sizes of 512 are used to compute the final outputs. We use Pointrend [7] to predict 2D segmentation masks.

During training, learning rate is set to 1×10^{-3} at first and reduced by a factor of 10 at the 50^{th} and 100^{th} epoch. When generating occlusion labels, learning rate is set to 1×10^{-5}. Training is performed with batch size of 32 for 150 epochs with Adam. We set $\lambda_\theta = 1$,$\lambda_\beta = 0.5$, $\lambda_{pos} = 10$, and $\lambda_{sil} = 0.1$ during training and $\lambda_\theta = \lambda_\beta = 0$ when generating occlusion labels in an self-supervised manner.

3 Additional Experiments

3.1 Model Performance and Efficiency

Table 1 compares the performance, running time, and memory of SOTA methods and HUMOR. Bottom-up methods are usually inferior in accuracy but superior in speed, especially when a large number of people appear. Our method surpasses both bottom-up and top-down methods in accuracy while remains high efficiency.

3.2 Broader Study of the Reasoning Module and the DSED Network on CrowdPose

CrowdPose is a more challenging 2D dataset compared with COCO. It features crowded scenes and contains more images with severe occlusions. We train all models on the training set of CrowdPose. Results are reported in Table 2. Compared with the original HigherHRNet, the DSED-based reasoning module

Table 2: **Broader study on 2D human pose.** Results are reported on CrowdPose. Reason stands for the reasoning module, Hg for the hourglass model. Our reasoning module brings more improvements in crowded scenes with occlusions, demonstrating the effectiveness of explicit occlusion reasoning.

		AP	AP^{50}	AP^{75}	AP^{E}	AP^{M}	AP^{H}
single-scale	HrHRNet-W48 [2]	65.9	86.4	70.6	73.3	66.5	57.9
	+ Reason (Hg)	64.6	85.0	69.8	73.4	65.0	55.8
	+ Reason (DSED)	70.3 (+4.4)	88.0 (+1.6)	76.5(+5.9)	78.2 (+4.9)	71.0 (+4.5)	61.3 (+3.4)
multi-scale	HrHRNet-W48 [2]	67.6	87.4	72.6	75.8	68.1	58.9
	+ Reason (Hg)	66.1	85.9	71.6	75.4	66.6	57.1
	+ Reason (DSED)	71.4 (+3.8)	88.7 (+1.3)	77.1 (+4.5)	79.3 (+3.5)	72.2 (+4.1)	62.1 (+3.2)

Table 3: **Comparisons on Human3.6M.** For our method, no ground-truth bounding box information is provided. We yield clear performance improvements. MPJPE is used.

	Method	MPJPE
top down	Lcr-net++ [18]	63.5
	Moon *et al.* [16]	54.4
	HMOR [19]	<u>48.6</u>
bottom up	ORPM [14]	69.9
	XNect [13]	63.6
	SMAP [21]	54.1
	Ours	50.3
	Ours(w/ Synth)	**47.6**

improves the results by 4.4 AP with single-scale testing and 3.8 AP with multi-scale testing, which are more significant than on COCO. Our reasoning module brings more improvements in crowded scenes with occlusions.

3.3 Comparison on Human3.6M

Human3.6M is a single-person dataset. Following [4,17], subjects 1,5,6,7,8 are used for training, and 9 and 11 for testing. The synthetic data here is different from the previous one. We generate a single-person synthetic dataset for this experiment. Results are reported in Table 3. Note that those top-down methods are essentially performing single-person pose estimation in a given bounding box, thus are more suitable for this dataset.

3.4 GCN *vs*. MLP

Different from [21], we use GCN instead of MLP for RefineNet. Table 4 shows the difference between them. The performance they achieve is similar, but GCN is more stable and in our implementation, it takes less training time. In addition, we think GCN is able to encode the skeleton information, thus may have

Table 4: **Ablation study on RefineNet.**

	$PCK_{rel}\uparrow$	$MPJPE_{rel}\downarrow$	$PCK_{occ}\uparrow$	$MPJPE_{occ}\downarrow$	Training time on single GPU (hours)
RefineNet (MLP)	83.9	93.3	74.0	121.3	~ 7.5
RefineNet (GCN)	84.3	90.8	74.1	119.4	~ 5.6

Table 5: **Effect of layer-by-layer supervision.** All models are trained without synthetic data. S-1/2/3 represents the model trained with supervision on the first, second, and third convolutional blocks. Others are named in the same way.

	$\text{PCK}_{rel} \uparrow$	$\text{MPJPE}_{rel} \downarrow$	$\text{PCK}_{occ} \uparrow$	$\text{MPJPE}_{occ} \downarrow$
S-1/2/3/4	78.77	106.75	58.98	155.71
S-2/3/4	75.07	113.74	54.83	168.18
S-3/4	75.07	113.74	54.80	167.92
S-4	75.11	113.70	54.81	168.33
S-1/4	77.95	106.77	57.97	156.86

Table 6: **Ablation study on training with and without using images.**

	$\text{PCK}_{rel} \uparrow$	$\text{MPJPE}_{rel} \downarrow$	$\text{PCK}_{occ} \uparrow$	$\text{MPJPE}_{occ} \downarrow$
Det + Reason (w/o image)	78.77	106.75	58.98	153.12
Det + Reason (w/ image)	78.23	106.14	58.58	154.08

more potential for better performance (but is beyond the scope of this paper). Therefore, we use GCN as the RefineNet of our HUPOR.

3.5 Effect of layer-by-layer supervision

Considering the use of skip connection and two encoders, and this specific task, we provide layer-by-layer (block-by-block) supervision between these two encoders and name this model "deeply supervised" model. Table 5 shows the effect of layer-by-layer supervision. We find that providing supervision for each block, especially for the first block, is very important for this task.

3.6 Training reasoning module with images

Table 6 shows the results of training reasoning module with and without using images as input. We see similar performance under these two settings, which proves that the intermediate results of the detection module already provide visible cues similar to what our network can extract from images. However, training with images will limit the use of synthetic data and cause extra computational resources. Therefore, our reasoning module doesn't use images as input.

4 Additional Qualitative Results

4.1 Explanation of DSED

When training DSED, We minimize the MSE of the **outputs of all layers** between the teacher and the student encoder. To better explain DSED, we visualize the output features of these two encoders in Fig. 4. **Note that the teacher encoder we visualized here is only trained with occluded joints.**

4.2 Qualitative Results of HUPOR on Images from MuPoTS-3D, 3DPW, and YouTube

Qualitative results of human pose estimation on images from MuPoTS-3D, 3DPW, and YouTube videos can be found in Fig. 5. Compared with the current SOTA method (*i.e.*, SMAP), HUPOR is more robust under severe occlusions and truncation, and generalizes much better to in-the-wild images. HUPOR works well in real-world applications with occlusions.

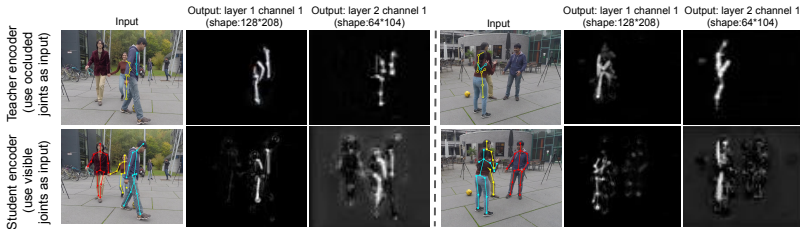


Fig. 4: **Output features of the teacher and the student encoder in DSED.** We visualize one channel of the first two layers (256 and 512 channels). Occluded keypoints are visualized as empty circles. During training, the teacher encoder takes occluded joints and learns features to reconstruct them and infers nearby skeletons. The student uses visible joints as input. It extracts useful information for occlusion reasoning from visible joints, and tries to infer occluded joints. During inference, only the student is used. Both encoders try to output the same features: features that are the best for reconstructing occluded joints and meanwhile, the easiest to be extracted from visible cues. Here input images are selected from test set, and the teacher is only used for visualization.

4.3 Qualitative Results of SSF

Qualitative results of human mesh reconstruction on MuPoTs-3D can be found in Fig. 6. Compared with the baseline method, SSF is more accurate and robust to images not seen before.

4.4 Visualization of Keypoint Detection and Reasoning

Fig. 7 provides more visualization results of the occluded joints reasoning process. For each image, we randomly select three occluded joints and visualize the outputs of the detection module and the DSED-based reasoning module. Our method detects the visibility of each keypoints and precisely localizes the position of visible joints, and more importantly, with the help of our reasoning module, it is able to reasonably infer occluded joints.

5 Limitations and Failure Cases

Fig. 8 shows some challenging scenarios. First, our method doesn't explicitly predict which occluded joints can be inferred and which cannot. Therefore, when severe occlusion or truncation occurs, HUPOR may give implausible estimates for those occluded joints. We think this problem can be solved by explicitly predicting which occluded joints can be inferred and regressing multiple plausible poses [5,9,1,20] for joints that cannot be inferred. The second problem our method suffers from is the noisy background. HUPOR may give false-positive predictions when handling images with very noisy backgrounds or with objects that look similar to a human (*e.g.* a doll).

Fig. 5: **Qualitative results of HUPOR on images from MuPoTS-3D, 3DPW, and YouTube.** From left to right: input image, (a) SMAP results, (b) our results. HUPOR is more robust under occlusions and truncation, and generalizes much better to in-the-wild images (YouTube).

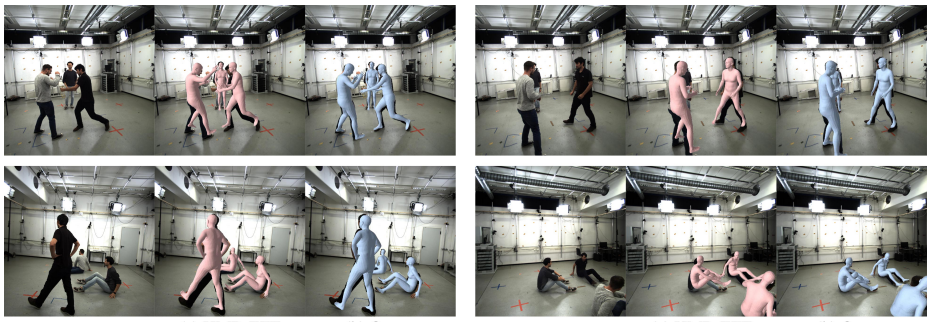


Fig. 6: **Qualitative results on human shape estimation.**

Fig. 7: **Keypoint detection and reasoning.** Images are cropped only for visualization. For each image, we randomly select three occluded joints and visualize the outputs of the detection module and the DSED-based reasoning module. Our method is able to reasonably infer occluded joints.

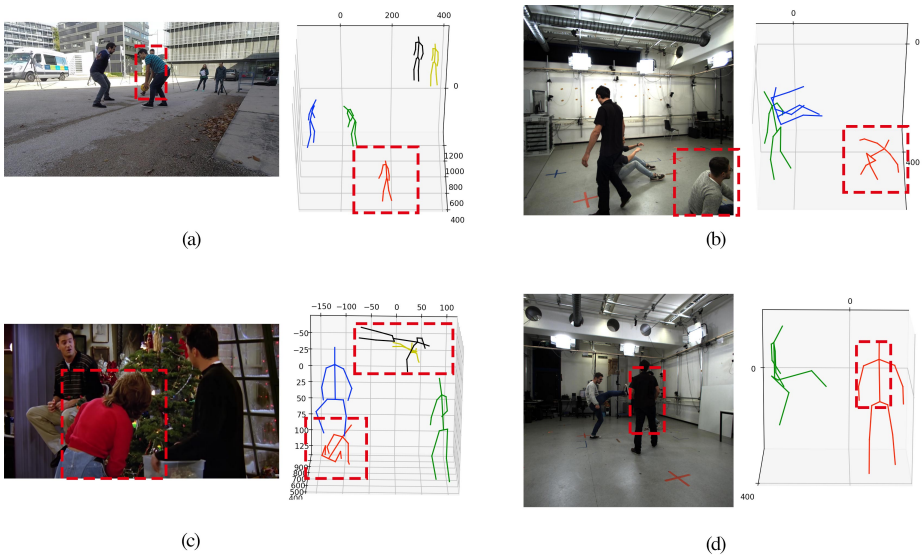


Fig. 8: **Failure cases.** HUPOR fails to produce fairly good estimate when handling severe occlusions (a,d), severe truncations (b,c), and very noise backgrounds (c).

References

1. Biggs, B., Ehrhadt, S., Joo, H., Graham, B., Vedaldi, A., Novotny, D.: 3d multi-bodies: Fitting sets of plausible 3d human models to ambiguous image data. Advances in Neural Information Processing Systems (2020) 7
2. Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T.S., Zhang, L.: Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5386–5395 (2020) 2, 3, 5
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 4
4. Hossain, M.R.I., Little, J.J.: Exploiting temporal information for 3d human pose estimation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 68–84 (2018) 5
5. Jahangiri, E., Yuille, A.L.: Generating multiple diverse hypotheses for human 3d pose consistent with 2d joint detections. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 805–814 (2017) 7
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 3
7. Kirillov, A., Wu, Y., He, K., Girshick, R.: Pointrend: Image segmentation as rendering. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9799–9808 (2020) 4
8. Kreiss, S., Bertoni, L., Alahi, A.: Pifpaf: Composite fields for human pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11977–11986 (2019) 2
9. Li, C., Lee, G.H.: Generating multiple hypotheses for 3d human pose estimation with mixture density network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9887–9895 (2019) 7
10. Li, J., Xu, C., Chen, Z., Bian, S., Yang, L., Lu, C.: Hybrik: A hybrid analytical-neural inverse kinematics solution for 3d human pose and shape estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3383–3393 (2021) 4
11. Li, W., Wang, Z., Yin, B., Peng, Q., Du, Y., Xiao, T., Yu, G., Lu, H., Wei, Y., Sun, J.: Rethinking on multi-stage networks for human pose estimation. arXiv preprint arXiv:1901.00148 (2019) 3
12. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014) 3
13. Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Elgharib, M., Fua, P., Seidel, H.P., Rhodin, H., Pons-Moll, G., Theobalt, C.: Xnect: Real-time multi-person 3d human pose estimation with a single rgb camera. ACM Transactions on Graphics (TOG) (2020) 5
14. Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Sridhar, S., Pons-Moll, G., Theobalt, C.: Single-shot multi-person 3d pose estimation from monocular rgb. In: 2018 International Conference on 3D Vision (3DV). pp. 120–130. IEEE (2018) 3, 5
15. Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.P., Xu, W., Casas, D., Theobalt, C.: Vnect: Real-time 3d human pose estimation with a single rgb camera. ACM Transactions on Graphics (TOG) **36**(4), 1–14 (2017) 3

16. Moon, G., Chang, J.Y., Lee, K.M.: Camera distance-aware top-down approach for 3d multi-person pose estimation from a single rgb image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10133–10142 (2019) 4, 5
17. Pavllo, D., Feichtenhofer, C., Grangier, D., Auli, M.: 3d human pose estimation in video with temporal convolutions and semi-supervised training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7753–7762 (2019) 5
18. Rogez, G., Weinzaepfel, P., Schmid, C.: Lcr-net++: Multi-person 2d and 3d pose detection in natural images. IEEE transactions on pattern analysis and machine intelligence **42**(5), 1146–1161 (2019) 5
19. Wang, C., Li, J., Liu, W., Qian, C., Lu, C.: Hmor: Hierarchical multi-person ordinal relations for monocular multi-person 3d pose estimation. In: European Conference on Computer Vision. pp. 242–259. Springer (2020) 5
20. Wehrbein, T., Rudolph, M., Rosenhahn, B., Wandt, B.: Probabilistic monocular 3d human pose estimation with normalizing flows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11199–11208 (2021) 7
21. Zhen, J., Fang, Q., Sun, J., Liu, W., Jiang, W., Bao, H., Zhou, X.: Smap: Single-shot multi-person absolute 3d pose estimation. In: European Conference on Computer Vision. pp. 550–566. Springer (2020) 3, 4, 5