Occluded Human Mesh Recovery

1. Occlusion Analysis

Our work is the first to propose a top-down method for multi-human occlusion – prior works in this setting are primarily bottom-up. As can be seen from the results in the main paper, our approach combining local+global centermaps with context normalization is a simple but powerful change and outperforms baselines by great margins.

In this section, following [5], we report performance under person and object occlusions on 3DPW-OCC and 3DOH datasets in Tab. 1. We also compare OCHMR against two additional baselines PARE [5] and HMR-EFT [3] on these datasets. OCHMR improves performance under person occlusion outperforming the prior art. Note, OCHMR suffers from object occlusion as the body centermaps do not encode person-object context.

Method	**Person Occlusion**		**Object Occlusion**	
	3DPW-PC ↓	OCHuman ↑	3DPW-OCC ↓	3DOH ↓
SPIN [6]	128.4	16.9	95.6	104.3
ROMP [13]	119.7	15.6	-	-
HMR-EFT (with pose) [3]	116.4	20.8	94.4	75.2
PARE [5]	122.5	18.0	**90.5**	**63.3**
OCHMR (Ours)	**112.2**	**37.7**	93.5	98.6

Table 1. We report MPJPE/AP of methods under person and object occlusion. Evaluations are done using ground-truth boxes. HMR-EFT uses AlphaPose for 2D pose for 3D mesh fitting.

2. Evaluation on CMU-Panoptic Dataset

We perform evaluations on the CMU-Panoptic dataset [4] computing the MPJPE of the 3D keypoints of the mesh recovered by OCHMR with respect to other baselines in Tab. 2. We observe that OCHMR outperforms bottom-up ROMP [13] on social activities with high bbox overlap.

Method	**Haggling**	**Mafia**	**Ultim.**	**Pizza**	**Mean**
CRMH [2]	129.6	133.5	153.0	156.7	143.2
ROMP (bottom-up) [13]	**111.8**	129.0	148.5	**149.1**	134.6
SPIN [6]	124.3	132.4	150.4	153.5	140.2
OCHMR (Ours)	115.5	**123.7**	**142.6**	150.6	**133.1**

Table 2. We report MPJPE on the CMU-Panoptic benchmark. All methods are trained using same data for a fair comparison.

3. Implementation Details

We use ResNet-50 [1] backbone for our implementation. For training speedup, we use distributed training using PyTorch [10] using 8 GeForce RTX 2080 GPUs. The learning rate is set to $5e-5$ and we train for 100 epochs using SGD optimizer. Similar to [6], we resize the input image to 224×224 and use a batch size of 64. While training, we use data augmentations like rotation, scaling, brightness and contrast changes. Further, we do not use SMPLify [9], static-fits or any pseudo ground-truth when training. The $2D$ and $3D$ loss weights are set similar to [6]. The context estimator F follows the HRNet-W32 [12] backbone and is trained separately on the COCO [8] dataset. All the hyper-parameters are set similar to the official implementation of HRNet [12]. We use off-shelf bounding box detector in the form of FasterRCNN [11] from the *detectron2*[1] codebase.

[1] https://github.com/facebookresearch/detectron2

4. Context Normalization (CoNorm) Block Code

In this section, we describe the code of CoNorm Block in `PyTorch`. The code in Listing. 1 outlines the details of functions $\mathbf{\Phi}_{\text{latent}}$, $\mathbf{\Phi}_{\text{scale}}$ and $\mathbf{\Phi}_{\text{bias}}$.

```
class CoNorm(nn.Module):

    def __init__(self, num_channels, hidden_channels=128):
        """
            CoNorm Block for Occluded Human Mesh Recovery
            num_channels: number of channels in the intermediate feature X
            hidden_channels: K, dimensionality of the latent space of the CoNorm block
        """
        super(CoNorm, self).__init__()

        self.phi_latent = nn.Sequential(
                nn.Conv2d(2, hidden_channels, kernel_size=3, padding=1),
                nn.ReLU()
            )
        self.phi_scale = nn.Conv2d(hidden_channels, num_channels, kernel_size=3, padding=1)
        self.phi_bias = nn.Conv2d(hidden_channels, num_channels, kernel_size=3, padding=1)
        return

    def forward(self, X, context):
        """
            X: intemediate feature of the segmentation backbone
            context: local and global centermap concatenated channelwise
        """
        context = F.interpolate(context, size=X.size()[2:], mode='bilinear', align_corners=False)

        lambda_ = self.phi_latent(context)
        scale = self.phi_scale(lambda_)
        bias = self.phi_bias(lambda_)

        X_prime = X * scale + bias

        return X_prime
```

Listing 1. Code for CoNorm block.

5. CoNorm Block Placement

We analyse the effect of adding CoNorm blocks at various depths in the OCHMR backbone in Tab. 3. Specifically, we investigate insertion after each of the four ResNet blocks for context normalization denoted by : Depth 1*(early)*, Depth 2, Depth 3, Depth 4(*late*). Empirically, inserting after at Depth 2 achieved the best performance with MPJPE of 114.0mm an improvement of 14.4mm over baseline. This is helpful when designing parameter efficient conditioning models where injection of information at a particular depth is imporant. This flexibility is provided by the CoNorm blocks.

	SPIN	Depth 1	Depth 2	Depth 3	Depth 4
MPJPE ↓	128.4	114.8	**114.0**	115.6	118.4
PMPJPE ↓	82.1	76.7	**76.1**	78.9	80.0

Table 3. Comparison of insertion of a single CoNorm block at various depths in the backbone. Context based feature normalization achieved best performance after the second ResNet block.

6. Robustness to Human Detector Outputs

The performance of top-down methods is often dependent on the quality of the detected bounding boxes. We analyse the robustness of SPIN and OCHMR with varying detector confidence in Tab. 4 on the COCO [8], CrowdPose [7], OCHuman [14] datasets using FasterRCNN [11].

OCHMR outperforms SPIN at all confidence thresholds, especially at stricter confidence thresholds on all datasets. This is consistent with our gains shown with ground-truth bounding boxes in the main paper.

Min. BB	**COCO**			**CrowdPose**			**OCHuman**		
Confid.	#boxes	SPIN(AP)	OCHMR(AP)	#boxes	SPIN(AP)	OCHMR(AP)	#boxes	SPIN(AP)	OCHMR(AP)
0.0	104125	11.3	15.3	27480	14.8	21.4	30637	11.2	24.8
0.1	44346	10.7	14.6	21237	13.4	20.1	22247	10.6	23.5
0.2	31748	10.4	14.3	16594	13.0	19.5	16273	10.4	22.8
0.3	24848	9.4	14.1	14358	12.5	19.0	13603	9.5	21.8
0.4	17534	8.7	13.6	12968	12.1	18.6	11944	9.2	21.5
0.5	14381	7.9	13.2	11869	11.5	18.5	10654	8.6	21.4
0.6	12794	7.5	12.6	10868	11.2	18.0	9626	8.1	21.3
0.7	11712	7.1	12.4	9944	11.2	17.2	8699	7.6	21.3
0.8	10631	6.7	12.4	8964	10.6	16.4	7768	7.3	20.8
0.9	9053	6.5	12.3	7780	10.1	15.9	6644	7.0	20.6
0.99	5274	5.4	12.0	4659	9.6	15.7	4416	6.2	20.0

Table 4. Variation of performance of SPIN and OCHMR with respect to confidence score across datasets using Faster-RCNN bounding boxes.

7. Qualitative Results

We provide additional interesting qualitative results under extreme crowding and occlusion in Fig. 1, Fig. 2, and Fig. 3. We specifically focus on sport related activities. In contrast to bottom-up approach like ROMP, top-down OCHMR successfully recovers mesh of all humans present in the image under crowding.

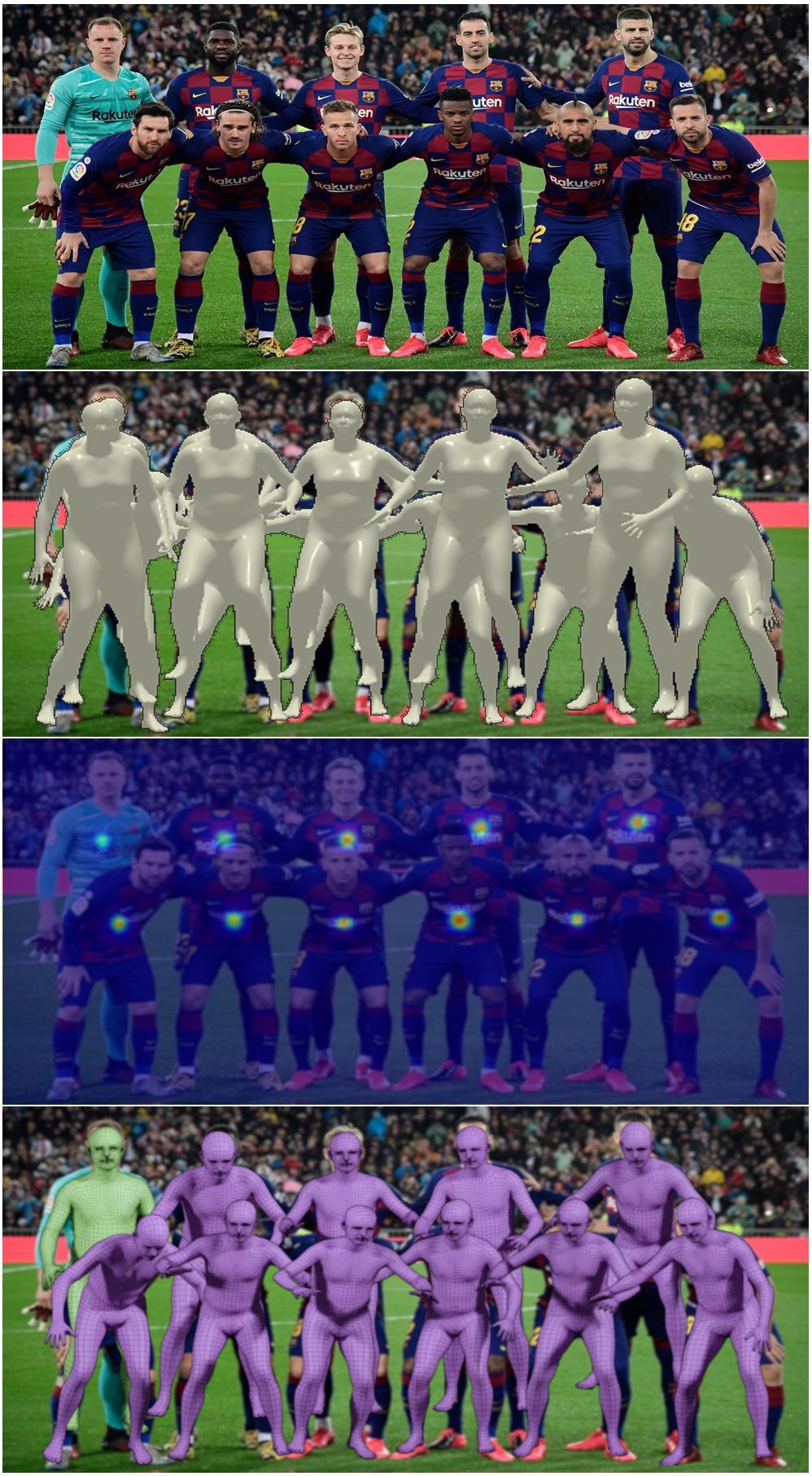

Figure 1. Qualitative results under challenging depth-ordering scenarios. Each image (top to bottom) shows RGB image, ROMP [13] predictions, OCHMR global-centermap predictions and OCHMR mesh predictions.

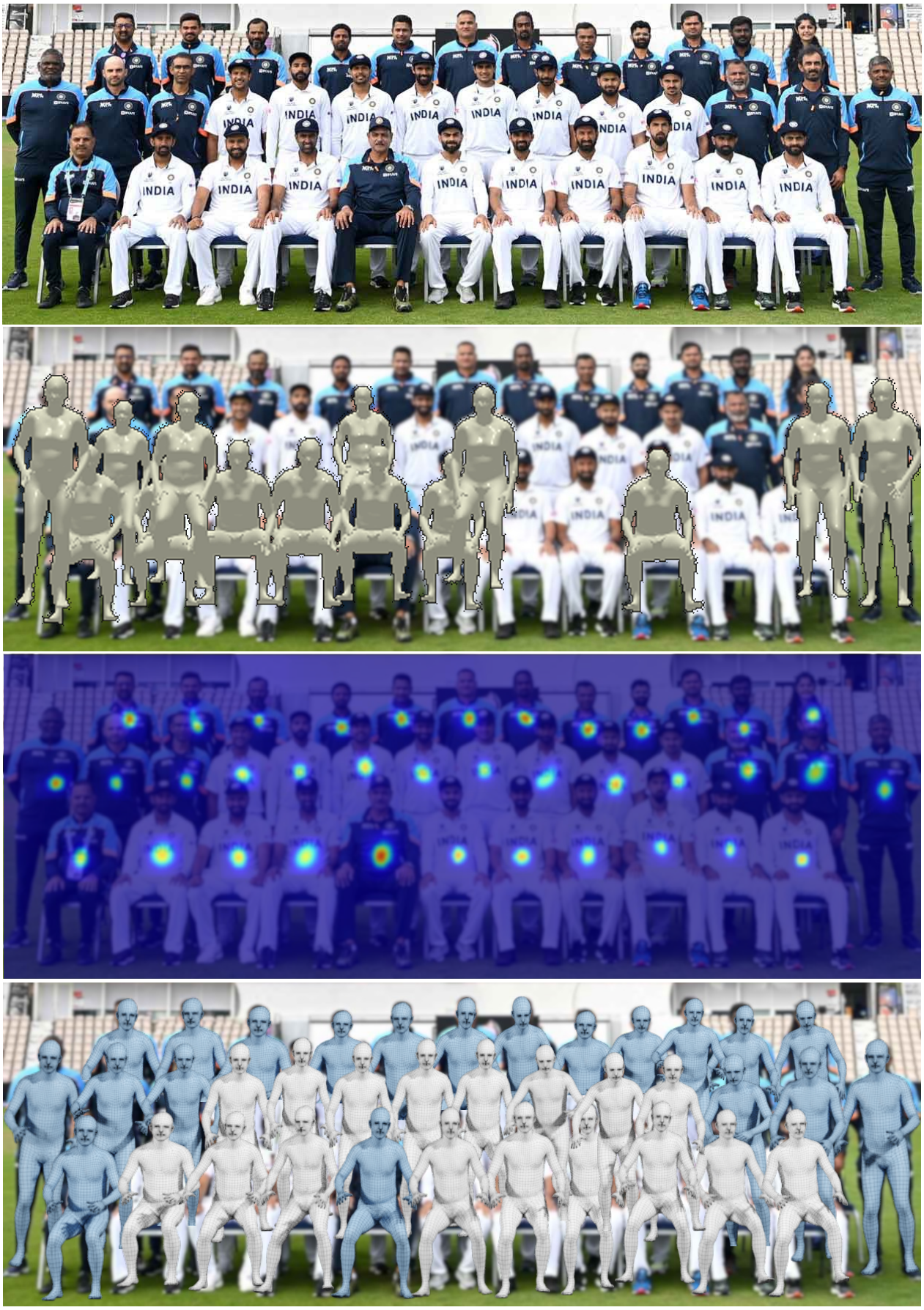

Figure 2. Qualitative results under severe crowding. Each image (top to bottom) shows RGB image, ROMP [13] predictions, OCHMR global-centermap predictions and OCHMR mesh predictions. ROMP fails to recover mesh for prominent humans in the image even after setting the confidence threshold to as low as 0.15. OCHMR recovers accurate meshes for all humans in the image.

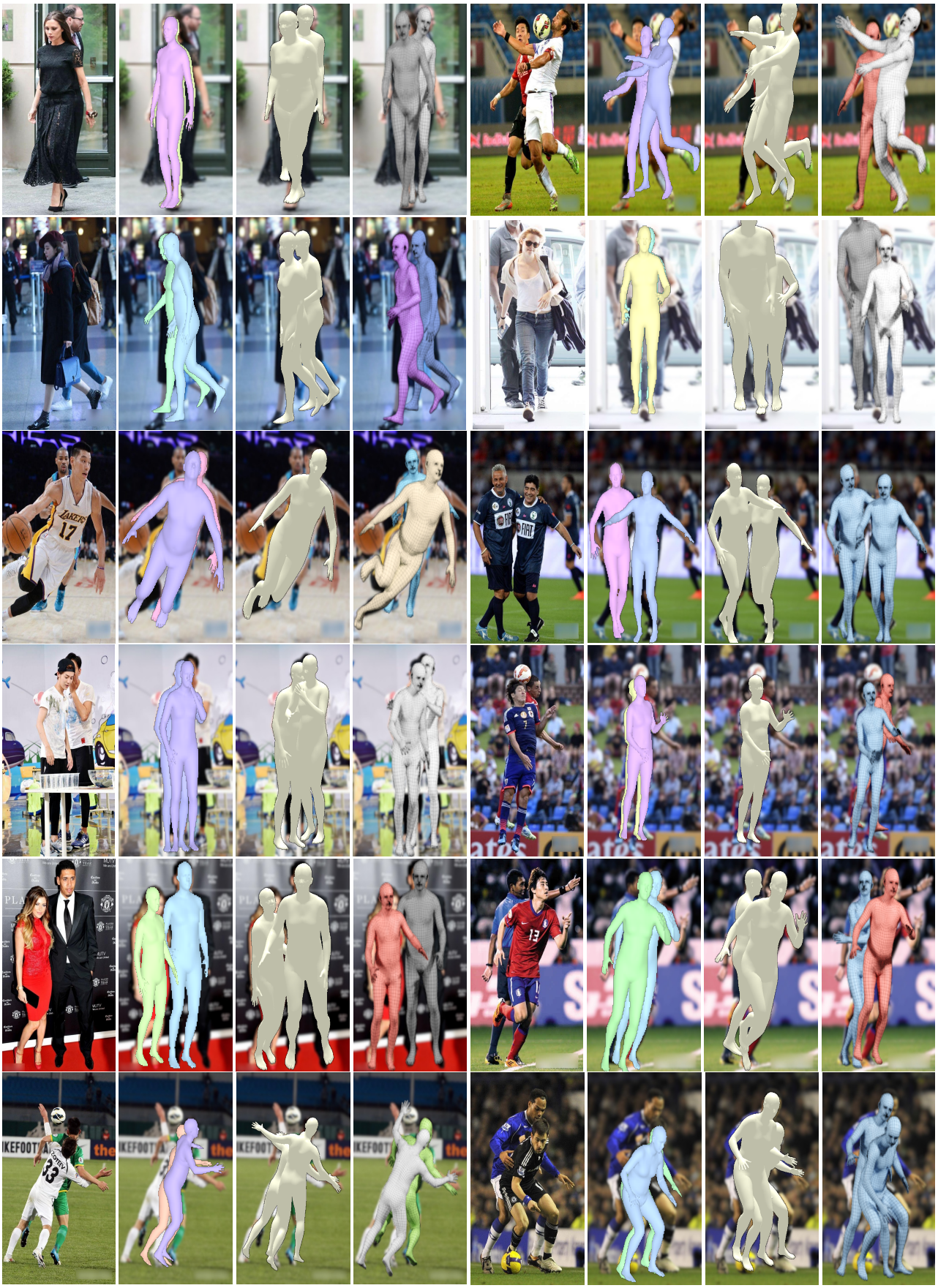

Figure 3. More qualitative results on the OCHuman dataset. Each image (left to right) shows RGB image, SPIN [6] predictions, ROMP [13] predictions and OCHMR predictions. OCHMR outputs pose consistent meshes under severe person-person occlusion.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[2] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2020. 1

[3] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human pose fitting towards in-the-wild 3d human pose estimation. In *3DV*, 2020. 1

[4] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Scott Godisart, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social interaction capture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 1

[5] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. *arXiv preprint arXiv:2104.08527*, 2021. 1

[6] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2252–2261, 2019. 1, 6

[7] 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. 3

[8] 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. 1, 3

[9] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 1

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 1

[11] 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*, 28:91–99, 2015. 1, 3

[12] 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

[13] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei. Monocular, one-stage, regression of multiple 3d people. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11179–11188, 2021. 1, 4, 5, 6

[14] 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. 3