FeatER: An Efficient Network for Human Reconstruction via Feature Map-Based TransformER — Supplementary Material

Ce Zheng[1], Matias Mendieta[1], Taojiannan Yang[1], Guo-Jun Qi[2,3], Chen Chen[1]
[1]Center for Research in Computer Vision, University of Central Florida
[2] OPPO Seattle Research Center, USA [3] Westlake University
{cezheng,mendieta,taoyang1122}@knights.ucf.edu; guojunq@gmail.com; chen.chen@crcv.ucf.edu

The supplementary material is organized into the following sections:

- Section A: Details of Complexity Comparison.
- Section B: Effectiveness of FeatER by Feature Maps Visualization.
- Section C: Effectiveness of Using the Feature Map Reconstruction Module.
- Section D: 2D-3D Lifting Module.
- Section E: Loss Function.
- Section F: More Qualitative Results.
- Section G: Broader Impact and Limitations.

A. Details of Complexity Comparison

In Table 1, we list the layer-by-layer comparison between one vanilla transformer block and one FeatER block. The shape of a stack of feature maps is $[n, h, w]$, where n is the number of feature maps, h and w is the height and width of the feature maps, respectively. If $h = w = 64$, the embedding dimension of d would be $d = hw = 4096$ without discarding any information. Since d is much larger than n, the computational complexity of one vanilla transformer block and one FeatER block can be written as $\mathcal{O}(d^2)$ and $\mathcal{O}(d^{3/2})$, respectively.

To be more specific, let there be a stack of 32 feature maps with the dimension of $[32, 64, 64]$. One vanilla transformer block requires 4.3G MACs when the embedding dimension is $d = 64 \times 64 = 4096$ (i.e., flattening the spatial dimension). Even if we further reduce the embedding dimension to $d = 1024$, it still needs 0.27G MACs. However, given feature maps $[32, 64, 64]$, FeatER only requires 0.09G MACs, which significantly reduces the computational cost.

B. Effectiveness of FeatER by Feature Maps Visualization

We visualize the coarse feature maps (extracted by CNN backbone) and the refined feature maps (refined by FeatER) in Fig. 1. These examples demonstrate that our proposed feature map-based transformer (FeatER) blocks can successfully refine the coarse feature maps by predicting more accurate joint locations, thereby improving the performance of human reconstruction tasks (2D HPE, 3D HPE, and HMR).

C. Effectiveness of Using the Feature Map Reconstruction Module

We compare the performance of our network with and without the feature map reconstruction module in Table 2. The performance is improved in all cases, including for the most challenging actions on the Human3.6M indoor dataset with heavy occlusions such as Photo, SitD (sitting down), and WalkD (walking with dog). The feature map reconstruction module effectively reduces the error by 4.4, 3.7, and 4.6 for these actions, respectively. Then, we compare the results on the in-the-wild 3DPW dataset, the MPJPE and MPVE also have decreased. Therefore, through this analysis, we validate the effectiveness of using the feature map reconstruction module.

Next, we investigate the best masking ratio in the feature map reconstruction module. We plot the relations between the error (MPJPE, PA-MPJPE, and MPVE) with the masking ratio in Fig. 2. We set the masking ratio to be 0.3 since it provides the best results on both Human3.6M and 3DPW datasets.

D. 2D-3D Lifting Module

The 2D-3D Lifting module is aimed to lift the 2D feature maps $[n, h, w]$ to 3D feature maps $[n, h, w, d]$. The intermediate 3D Pose can be obtained by a 3D pose head. The MLP

Table 1. The detailed complexity comparison between one vanilla transformer block and one FeatER block. We calculate their MACs based on the input and output with the corresponding operation.

Vanilla Transformer block					**FeatER block**				
Attention Layer:					**Attention Layer(AttentionW):**				
description	input	output	operation	MACs	description	input	output	operation	MACs
x to QKV	$x_{in}\ [n, d]$	$QKV\ [n, 3d]$	nn.Linear(d, 3d)	$3nd^2$	x to QKV	x^w_{in}: $[n, h, w]$	$QKV\ [n, h, 3w]$	nn.Linear(w, 3w)	$3nhw^2$
$a_1 = QK^T$	$Q[n, d]$, $K^T[d, n]$	$a_1[n, n]$	torch.matmul	n^2d	$a^w_1 = Q^wK^{wT}$	$Q^w[h, n, w]$,$K^{wT}[h, w, n]$	$a^w_1\ [h, n, n]$	torch.matmul	n^2hw
$x_{attn} = a_1V$	$a_1[n, n]$, $V[n, d]$	$x_{attn}[n, d]$	torch.matmul	n^2d	$x^w_{attn} = a^w_1V^w$	$a^w_1[h, n, n]$, $V[h, n, w]$	$x^w_{attn}[h, n, w]$	torch.matmul	n^2hw
					Attention Layer(AttentionH):				
					description	input	output	operation	MACs
					x to QKV	x^h_{in}: $[n, w, h]$	$QKV\ [n, w, 3h]$	nn.Linear(h, 3h)	$3nh^2w$
					$a^h_1 = Q^hK^{hT}$	$Q^h[w, n, h]$,$K^{hT}[w, h, n]$	$a^h_1\ [w, n, n]$	torch.matmul	n^2hw
					$x^h_{attn} = a^h_1V^h$	$a^h_1\ [w, n, n]$,$V^h\ [w, n, h]$	$x^w_{attn}[w, n, h]$	torch.matmul	n^2hw
Projection Layer					**Projection Layer**				
description	input	output	operation	MACs	description	input	output	operation	MACs
projection:	$x_{attn}[n, d]$	$x[n, d]$	nn.Linear(d, d)	nd^2	projection:	$x^{FM}_{attn}[n, h, w]$	$x[n, h, w]$	nn.Conv2d(n, n,1)	n^2hw
MLP Layer (mlp raito=2) in FFN:					**CONV Layer (conv raito=2) in FFN:**				
description	input	output	operation	MACs	description	input	output	operation	MACs
MLP	$x[n, d]$	$x_{hidden}[n, 2d]$	nn.Linear(d,2d)	$2nd^2$	CONV	$x[n, h, w]$	$x_{hidden}[2n, h, w]$	nn.Conv2d(n, 2n,1)	$2n^2hw$
MLP	$x_{hidden}[n, 2d]$	$x[n, d]$	nn.Linear(2d,d)	$2nd^2$	CONV	$x_{hidden}[2n, h, w]$	$x[n, h, w]$	nn.Conv2d(2n, n,1)	$2n^2hw$
Total: $8nd^2 + 2n^2d$					**Total:** $3nhw(h + w) + 9n^2(wh)$				
					Total: $6nd^{3/2} + 9n^2d$ when $w * h = d$ and $w = h$				

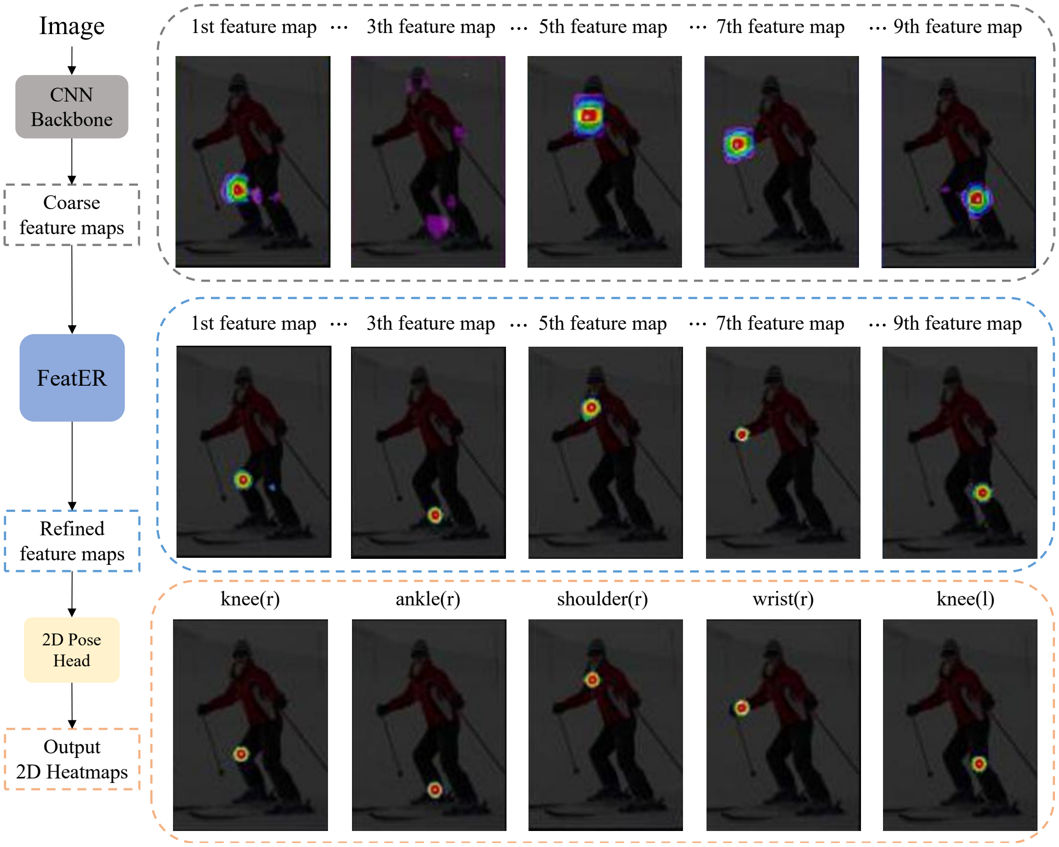
Image
CNN Backbone
Coarse feature maps
FeatER
Refined feature maps
2D Pose Head
Output 2D Heatmaps
1st feature map ··· 3th feature map ··· 5th feature map ··· 7th feature map ··· 9th feature map
1st feature map ··· 3th feature map ··· 5th feature map ··· 7th feature map ··· 9th feature map
knee(r)
ankle(r)
shoulder(r)
wrist(r)
knee(l)

Figure 1. Visualization of coarse feature maps (extracted by CNN backbone) and refined feature maps (refined by FeatER).

head outputs the parameters for the mesh regressor. The architecture of the 2D-3D Lifting Module is shown in Fig. 3.

E. Loss Function

2D HPE

We first train our FeatER on COCO dataset for the 2D HPE task. Following [3, 7], we apply the Mean

Table 2. Ablation study on the effectiveness of using our feature map reconstruction module on Human3.6M. 'FM-Rec' means Feature Map Reconstruction Module and 'Δ' denotes the performance improvement.

	Human3.6M										3DPW	
	MPJPE ↓										MPJPE ↓	MPVE ↓
actions	Dire.	Eat	Phone	Photo	Pose	Purch.	SitD.	WalkD.	Smoke	Avg.	Avg.	Avg.
w/o FM-Rec	50.1	49.5	56.8	60.0	46.3	51.0	69.4	57.8	52.4	53.3	89.9	106.9
with FM-Rec	46.3	45.7	54.7	55.6	43.0	47.2	65.7	53.2	49.6	49.9	88.4	105.6
Δ	3.8	3.8	2.1	4.4	3.3	3.8	3.7	4.6	2.8	3.4	1.5	1.3

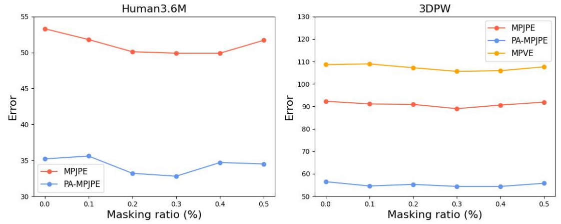

Figure 2. Evaluation of different masking ratios in the feature map reconstruction module.

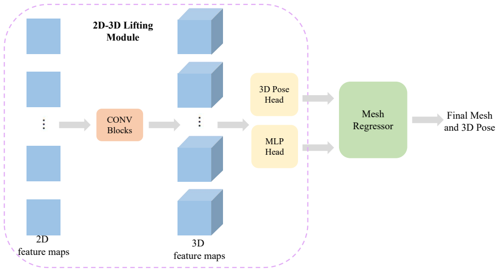

Figure 3. The architecture of the 2D-3D Lifting Module

Squared Loss (MSE) between the predicted heatmaps (HM) $HM \in \mathbb{R}^{K \times h \times w}$ and the ground truth 2D pose $HM^{GT} \in \mathbb{R}^{K \times h \times w}$, where K is the number of joints, h and w are the height and width of heatmaps, respectively. When the input image is 256×192 and the number of joints is $K = 17$, the heatmap size would be $w = 64$, and $h = 48$, respectively. The MSE for the 2D pose is defined as follows:

$$\mathcal{L}_{2D-Pose} = \|HM - HM^{GT}\|^2 \tag{1}$$

3D HPE and HMR

We apply an $L1$ loss between the predicted 3D pose $J \in \mathbb{R}^{K \times 3}$ and the ground truth 3D pose $J_{GT} \in \mathbb{R}^{K \times 3}$ following [1,2,4]. K is the number of joints.

$$\mathcal{L}_{3D-Pose} = \frac{1}{K} \sum_{i=1}^{K} \|J_i - J_i^{GT}\|_1 \tag{2}$$

Following [2], we use the SMPL [6] model to output human mesh, which is obtained by fitting the 3D pose J, the shape parameter β, and the rotation parameter θ into the SMPL model. We supervise the shape and rotation parameters by applying the $L2$ loss following [6]. The reconstruction loss $\mathcal{L}_{rec}$ is the Mean Square Error (MSE) between the target feature maps and reconstructed feature maps. The overall loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{overall} = \mathcal{L}_{3D-Pose} &+ w_1\|\beta - \beta^{GT}\| \\ &+ w_2\|\theta - \theta^{GT}\| + w_3\mathcal{L}_{rec} \end{aligned} \tag{3}$$

where $w_1 = 0.01$, $w_2 = 0.01$ and $w_3 = 0.005$ are the weights for the loss terms.

F. More Qualitative Results

2D Heatmap and Human Mesh Reconstruction (HMR) Visualization

Fig. 4 provides visualization of 17 heatmaps (COCO [5] 17 joints format) and the predicted 2D poses of the input images. The visualization of Human3.6M and 3DPW dataset are shown in Figs. 6. Figs. 5 and 7 show the HMR visualization of FeatER on several in-the-wild images from the COCO [5] dataset. FeatER can estimate accurate human meshes of the given images with regular human articulation in Fig. 5. For some very challenging cases, as shown in Fig. 7, FeatER can still output reliable human meshes.

When comparing with the state-of-the-art HMR method METRO [4], FeatER clearly outperforms METRO with only 5% of Params and 16% of MACs on these in-the-wild images (taken from the COCO [5] dataset) as depicted in Fig. 8, demonstrating the superiority (in terms of both accuracy and efficiency) of the proposed FeatER method for practical applications.

Inaccurate and Failure Cases

Although FeatER can estimate human mesh quite well as demonstrated in Figs. 5 and 7, there are still some inaccurate and failure cases. As presented in Fig. 9 left, the red circle indicates the inaccurate mesh part due to heavy occlusion. The proposed Feature Map Reconstruction Module is not enough to tackle this issue with limited training data. For more complex human body articulation in Fig. 9 right, FeatER fails to estimate accurate human mesh. How to further improve the generalization of FeatER to in-the-wild images would be our future work.

G. Broader Impact and Limitations

We believe that FeatER will help to highlight model efficiency for the HMR task. With significantly reduced computational and memory complexity than SOTA approaches,

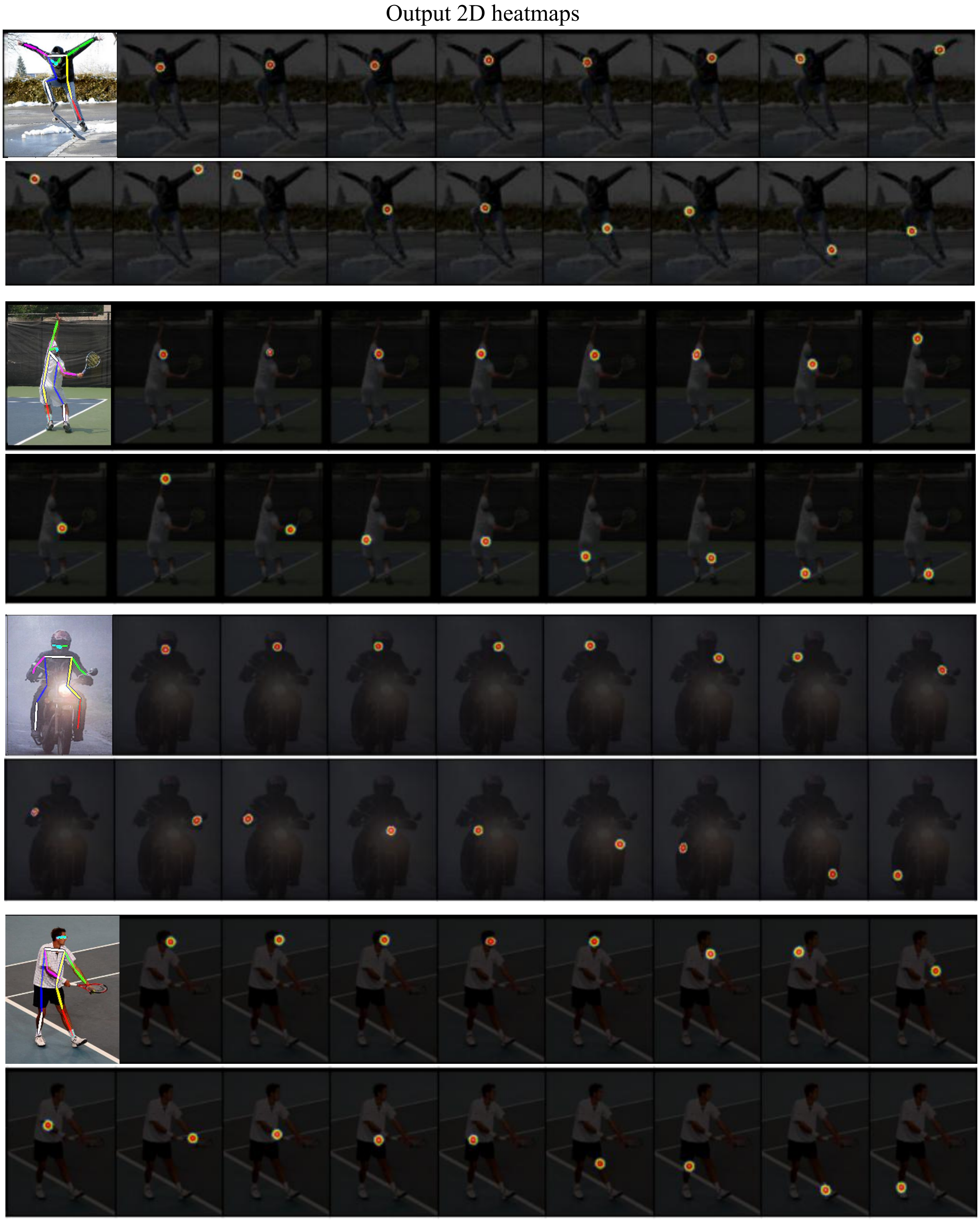

Figure 4. 2D heatmaps visualization of the proposed FeatER. Images are taken from the COCO validation set [5].

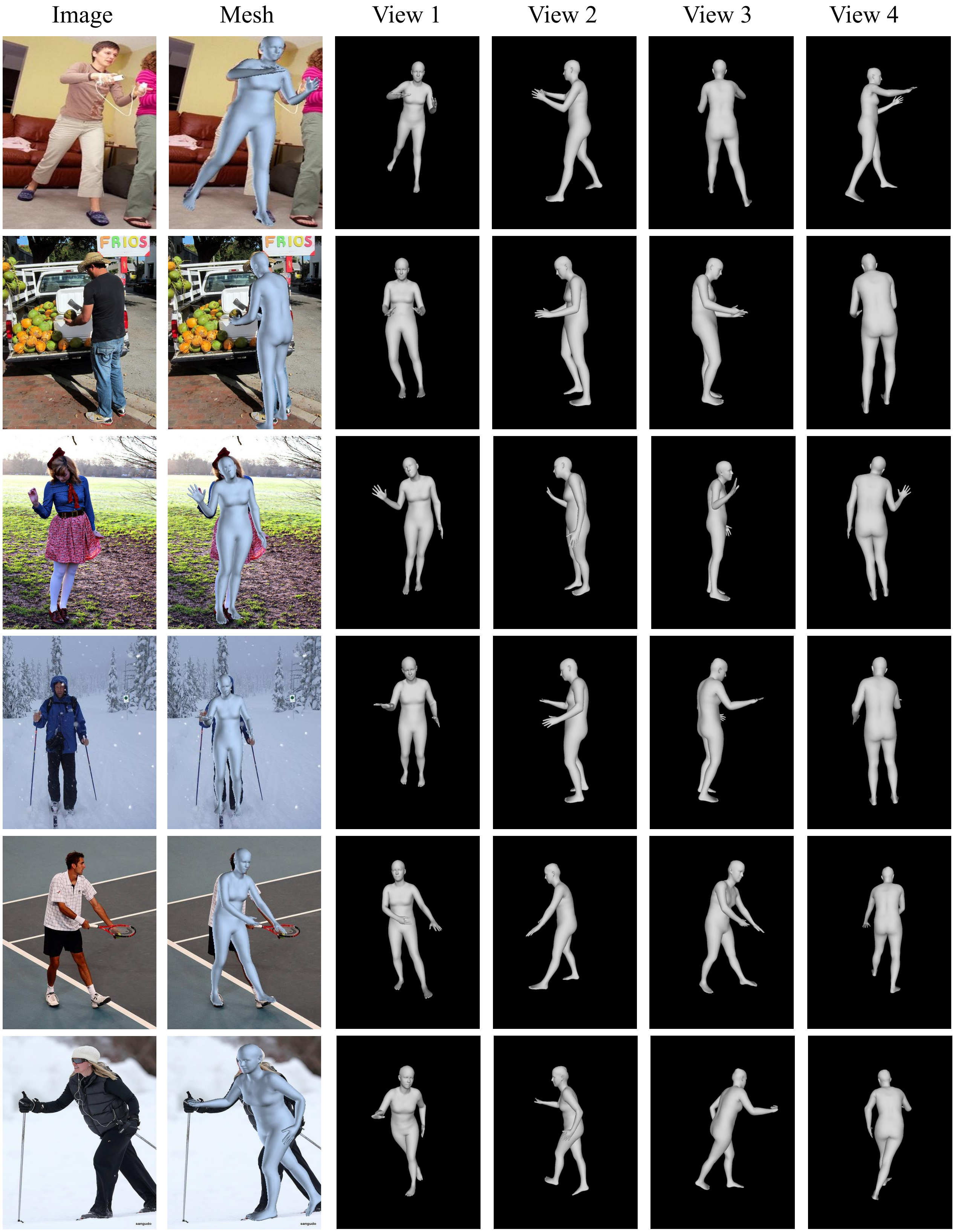

Figure 5. Mesh reconstruction qualitative results of the proposed FeatER. Images are taken from the in-the-wild COCO [5] dataset.

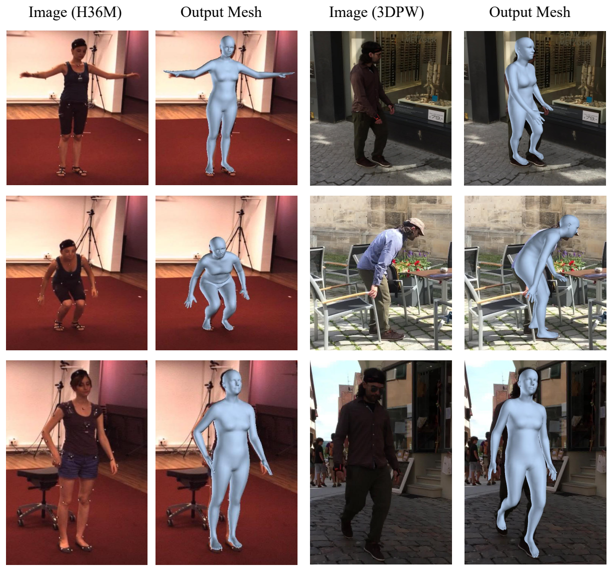

Figure 6. Mesh reconstruction qualitative results of the proposed FeatER. Images are taken from the Human3.6M dataset and 3DPW dataset.

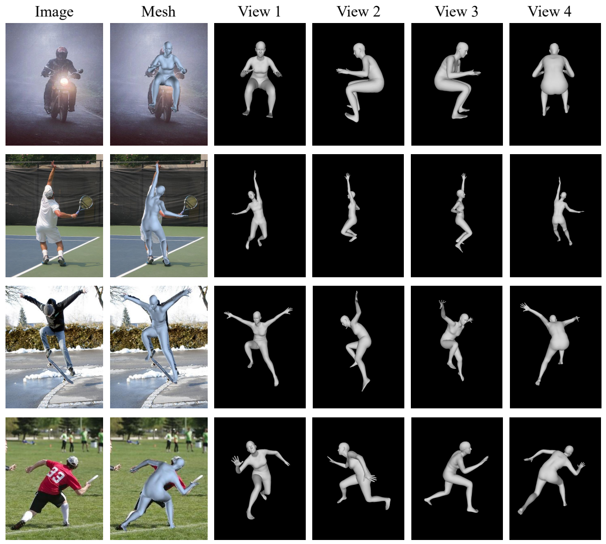

Figure 7. Mesh reconstruction qualitative results of the proposed FeatER for more challenging cases. Images are taken from the in-the-wild COCO [5] dataset.

FeatER can still outperform them, which is better appreciated by real-world applications like VR/AR, virtual try-on, and AI coaching.

A potential limitation of FeatER is that it can not perform well in some specific scenarios such as crowded scenes. We leave this issue for future study.

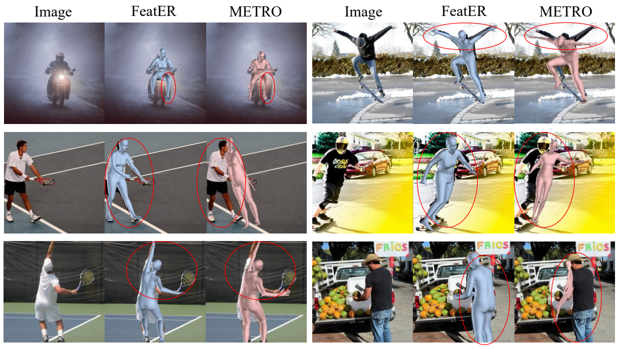

Figure 8. Qualitative comparison with the state-of-the-art HMR method METRO [4]. Images are taken from the in-the-wild COCO [5] dataset. The red circles highlight locations where FeatER is more accurate than METRO.

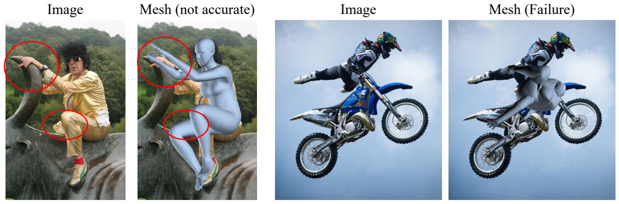

Figure 9. Left: Inaccurately estimated mesh due to heavy occlusion. Right: Failure estimated mesh due to complex human body articulation.

References

[1] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, 2020. 3

[2] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. 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*, pages 3383–3393, 2021. 3

[3] 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

[4] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1954–1963, 2021. 3, 6

[5] 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. 3, 4, 5, 6

[6] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM TOG*, 2015. 3

[7] 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. 2