FEATER: AN EFFICIENT NETWORK FOR HUMAN RECONSTRUCTION VIA FEATURE MAP-BASED TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, vision transformers have shown great success in a set of human reconstruction tasks such as 2D human pose estimation (2D HPE), 3D human pose estimation (3D HPE), and human mesh reconstruction (HMR) tasks. In these tasks, feature map representations of the human structural information are often extracted first from the image by a CNN (such as HRNet), and then further processed by transformer to predict the heatmaps (encodes each joint's location into a feature map with a Gaussian distribution) for HPE or HMR. However, existing transformer architectures are not able to process these feature map inputs directly, forcing an unnatural flattening of the location-sensitive human structural information. Furthermore, much of the performance benefit in recent HPE and HMR methods has come at the cost of ever-increasing computation and memory needs. Therefore, to simultaneously address these problems, we propose FeatER, a novel transformer design which preserves the inherent structure of feature map representations when modeling attention while reducing the memory and computational costs. Taking advantage of FeatER, we build an efficient network for a set of human reconstruction tasks including 2D HPE, 3D HPE, and HMR. A feature map reconstruction module is applied to improve the performance of the estimated human pose and mesh. Extensive experiments demonstrate the effectiveness of FeatER on various human pose and mesh datasets. For instance, FeatER outperforms the SOTA method MeshGraphormer by requiring 5% of Params (total parameters) and 16% of MACs (the Multiply–Accumulate Operations) on Human3.6M and 3DPW datasets. Code will be publicly available.

1 INTRODUCTION

Understanding human structure from monocular images is one of the fundamental topics in computer vision. The corresponding tasks of Human Pose Estimation (HPE) and Human Mesh Reconstruction (HMR) have received a growing interest from researchers, accelerating progress towards various applications such as VR/AR, virtual try-on, and AI coaching. However, HPE and HMR from a single image still remain challenging tasks due to depth ambiguity, occlusion, and complex human body articulation.

With the blooming of deep learning techniques, Convolutional Neural Network (CNN) Simonyan & Zisserman (2014); He et al. (2016); Sun et al. (2019) architectures have been extensively utilized in vision tasks and have achieved impressive performance. Most existing HPE and HMR models Sun et al. (2019); Khirodkar et al. (2022) utilize CNN-based architectures (such as ResNet He et al. (2016) and HRNet Sun et al. (2019)) to predict feature maps, which are supervised by the ground-truth 2D heatmap representation (encodes the position of each keypoint into a feature map with a Gaussian distribution) as shown in Fig. 1. This form of output representation and supervision can make the training process smoother, and therefore has become the *de facto* process in HPE's networks Cao et al. (2017); Sun et al. (2019); Zhang et al. (2020).

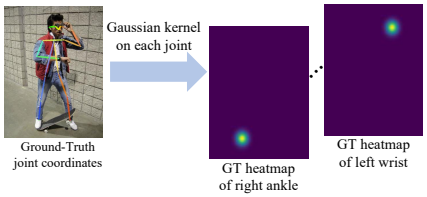

Figure 1: Generating heatmaps from joint coordinates.

Recently, the transformer architecture has been fruitfully adapted from the field of natural language processing (NLP) into computer vision, where it has enabled state-of-the-art performance in HPE and HMR tasks Yang et al. (2021); Li et al. (2021e); Zheng et al. (2021); Lin et al. (2021b;a). The transformer architecture demonstrates a strong ability to model global dependencies in comparison to CNNs via its self-attention mechanism. The long-range correlations between tokens can be captured, which is critical for modeling the dependencies of different human body parts in HPE and HMR tasks. Since feature maps concentrate on certain human body parts, we aim to utilize the transformer architecture to refine the coarse feature maps (extracted by a CNN backbone). After capturing the global correlations between human body parts, more accurate pose and mesh can be obtained.

However, inheriting from NLP where transformers embed each word to a feature vector, Vision Transformer architectures such as ViT Dosovitskiy et al. (2021) can only deal with the flattened features when modeling attention. This is less than ideal for preserving the structural context of the feature maps during the refinement stage (feature maps with the shape of $[n, h, w]$ needs to be flattened as $[n, d]$, where $d = h \times w$. Here n is the number of feature maps, h and w are height and width of each feature map, respectively). Furthermore, another issue is that the large embedding dimension brought about by the flattening process makes the transformer computational expensive. This is not friendly to the real-world applications of HPE and HMR, which often demand real-time processing capabilities on deployed devices (e.g. AR/VR headsets).

Therefore, in this paper, we propose a Feature map-based transformER (FeatER) architecture to properly refine the coarse feature maps through global correlations of structural information in a resource-friendly manner. Compared to the vanilla transformer architecture, FeatER has two advantages:

- First, FeatER preserves the feature map representation in the transformer encoder when modeling self-attention, which is naturally adherent with the HPE and HMR tasks. Rather than conducting the self-attention based on flattened features, FeatER ensures that the self-attention is conducted based on the original 2D feature maps, which is more structurally meaningful. To accomplish this, FeatER is designed with a novel dimensional decomposition strategy to handle the extracted stack of 2D feature maps.
- Second, this decompositional design simultaneously provides a significant reduction in computational cost compared with the vanilla transformer [1]. This makes FeatER more suitable for the needs of real-world applications.

Equipped with FeatER, we present an efficient framework for human representation tasks including 2D HPE, 3D HPE, and HMR. For the more challenging 3D HPE and HMR portion, a feature map reconstruction module is integrated into the framework. Here, a subset of feature maps are randomly masked and then reconstructed by FeatER, enabling more robust 3D pose and mesh predictions for in-the-wild inference. We conduct extensive experiments on human representation tasks, including 2D human pose estimation on COCO, 3D human pose estimation and human mesh reconstruction on Human3.6M and 3DPW datasets. Our method (FeatER) consistently outperforms SOTA methods on these tasks with significant computation and memory cost reduction (e.g. FeatER outperforms MeshGraphormer Lin et al. (2021a) with only requiring 5% of Params and 16% of MACs).

2 Related work

Since Vision Transformer (ViT) Dosovitskiy et al. (2021) introduced the transformer architecture to image classification with great success, it has also been shown to have enormous potential in various vision tasks such as object detection Misra et al. (2021); Liu et al. (2021), facial expression recognition Xue et al. (2021), and re-identification Li et al. (2021c); He et al. (2021). Since the related work of HPE and HMR is vast, we refer interested readers to the recent and comprehensive surveys: Chen et al. (2020b) for HPE and Tian et al. (2022) for HMR. In this section, we discuss the more relevant transformer-based approaches.

[1] For example, there are 32 feature maps with overall dimension $[32, 64, 64]$. For a vanilla transformer, without discarding information, the feature maps needs to be flattened into $[32, 4096]$. One vanilla transformer block requires 4.3G MACs. Even if we reduce the input size to $[32, 1024]$, it still requires 0.27G MACs. However, given the original input of $[32, 64, 64]$, FeatER only requires 0.09G MACs.

Transformers in HPE: HPE can be categorized into 2D HPE and 3D HPE based on 2D pose output or 3D pose output. Recently, several methods Yang et al. (2021); Li et al. (2021e;b) utilize transformers in 2D HPE. TransPose Yang et al. (2021) uses a transformer to capture the spatial relationships between keypoint joints. PRTR Li et al. (2021b) builds cascade transformers with encoders and decoders based on DETR Carion et al. (2020). Although achieving impressive performance, TransPose Yang et al. (2021) and PRTR Li et al. (2021b) suffer from heavy computational costs. HRFormer Yuan et al. (2021) integrates transformer blocks in the HRNet structure to output 2D human pose. TokenPose Li et al. (2021e) embeds each keypoint to a token for learning constraint relationships by transformers, but it is limited to the 2D HPE task. At the same time, PoseFormer Zheng et al. (2021) and Li et al. Li et al. (2022) first apply transformers in 3D HPE. MHFormer Li et al. (2021d) generates multiple plausible pose hypotheses using transformers to lift 2D pose input to 3D pose output. As 2D-3D lifting approaches, these methods Zheng et al. (2021); Li et al. (2022; 2021d) rely on the external 2D pose detector, which is not end-to-end. In contrast, our FeatER is an end-to-end network for the 2D HPE, 3D HPE, and HMR in a resource-friendly manner.

Transformers in HMR: THUNDR Zanfir et al. (2021) introduces a model-free transformer-based architecture for human mesh reconstruction. GTRS Zheng et al. (2022) proposes a graph transformer architecture with parallel design to estimate 3D human mesh only from detected 2D human pose. METRO Lin et al. (2021b) combines a CNN backbone with a transformer network to regress human mesh vertices directly for HMR. MeshGraphormer Lin et al. (2021a) further injects GCNs into the transformer encoder in METRO to improve the interactions among neighboring joints. Despite their excellent performance, METRO and MeshGraphormer still incur substantial memory and computational overhead.

Efficient Methods for HPE and HMR: The SOTA methods for HPE and HMR Lin et al. (2021b;a) mainly pursue higher accuracy without considering computation and memory cost. While less studied, model efficiency is also a key characteristic for HPE and HMR applications. Lite-HRNet Yu et al. (2021) applys the efficient shuffle block in ShuffleNet Zhang et al. (2018); Ma et al. (2018) to HRNet Sun et al. (2019), but it is only limited to 2D HPE. GTRS Zheng et al. (2022) is a lightweight pose-based method that can reconstruct human mesh from 2D human pose. However, to reduce the computation and memory cost, GTRS only uses 2D pose as input and therefore misses some information such as human shape. Thus, the performance is not comparable to the SOTA HMR methods Dwivedi et al. (2021); Lin et al. (2021a). Our FeatER is an efficient network that can outperform SOTA methods while reducing computation and memory costs significantly.

3 METHODOLOGY

3.1 OVERVIEW ARCHITECTURE

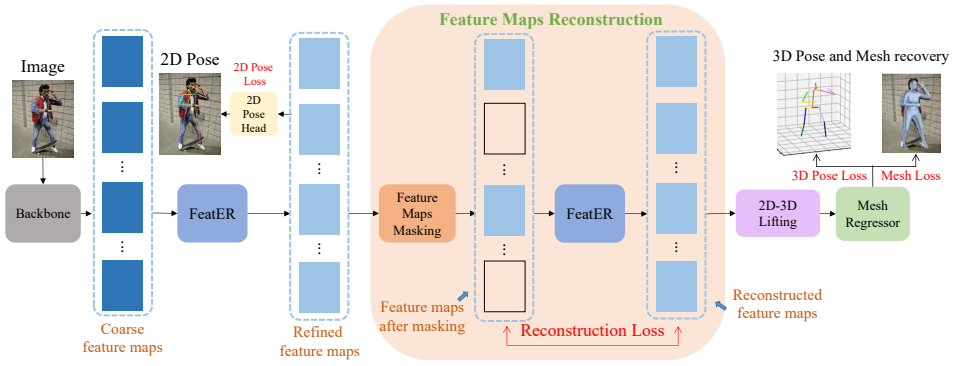

Figure 2: An overview of our proposed network for 2D HPE, 3D HPE, and HMR tasks. The coarse feature maps are extracted by the CNN backbone and refined by FeatER blocks. The 2D pose can be obtained by a 2D pose head. Then, we apply a feature map reconstruction module to improve the robustness of the predicted 3D pose and mesh. This is accomplished by randomly masking out some feature maps and utilizing FeatER blocks to reconstruct them. Next, we apply a 2D-3D lifting module which converts the 2D feature maps to 3D feature maps, and predicts the parameters for the mesh regressor. Finally, the mesh regressor outputs the 3D human pose and mesh.

As shown in Fig. 2, we propose a network for 2D HPE, 3D HPE and HMR tasks. Given an image, a CNN backbone is applied first to extract the coarse feature maps. Then, our proposed FeatER blocks further refine the feature maps by capturing the global correlations between them. Next, a 2D pose head is used to output the 2D pose. To improve the robustness of the estimated 3D pose and mesh, we apply a feature map reconstruction module with a masking strategy. Specifically, a subset of the feature maps are randomly masked with a fixed probability, and then FeatER blocks are tasked with reconstruction. Finally, a 2D-3D lifting module and a mesh regressor (we use the same regressor as in HybrIK Li et al. (2021a)) outputs the estimated 3D pose and mesh.

3.2 PRELIMINARIES OF VANILLA TRANSFORMER

The input of a vanilla transformer block is $X_{in} \in \mathbb{R}^{n \times d}$, where n is the number of patches and d is the embedding dimension. Vanilla transformer block is composed of the following operations, and is applied to capture global dependencies between patches via self-attention mechanism.

Multi-head Self-Attention Layer (MSA) is the core function to achieve self-attention modeling. After layer normalization, the input $X_{in} \in \mathbb{R}^{n \times d}$ is first mapped to three matrices: query matrix Q, key matrix K and value matrix V by three linear transformation:

$$Q = X_{in}W_Q, \quad K = X_{in}W_K, \quad V = X_{in}W_V. \tag{1}$$

where W_Q, W_K and $W_V \in \mathbb{R}^{d \times d}$.

The scaled dot product attention can be described as the following mapping function:

$$\text{Attention}(Q, K, V) = \text{Softmax}(QK^\top/\sqrt{d})V. \tag{2}$$

where $\frac{1}{\sqrt{d}}$ is the scaling factor for appropriate normalization to prevent extremely small gradients.

Next, the vanilla transformer block architecture consisting of MSA and feed-forward network (FFN) is shown in Fig. 3 (a). The block output $X_{out} \in \mathbb{R}^{n \times d}$ keeps the same size as the block input $X_{in} \in \mathbb{R}^{n \times d}$, and is represented as follows:

$$X_{attn} = \text{MSA}(Q, K, V) + X_{in} \tag{3}$$

$$X_{out} = \text{FFN}(X_{attn}) + X_{attn} \tag{4}$$

where $\text{MSA}(\cdot)$ represents the Multi-head Self-Attention block, and $\text{FFN}(\cdot)$ is a feed-forward network which consists of the multilayer perceptron (MLP) and normalization layer.

Thus, given a sequence of coarse 2D feature maps (FM) extracted by the CNN backbone $X_{in}^{FM} \in \mathbb{R}^{n \times h \times w}$, the vanilla transformer block needs to flatten the feature maps into $X_{in} \in \mathbb{R}^{n \times d}$, where $d = h \times w$. After the vanilla transformer block, the output $X_{out} \in \mathbb{R}^{n \times d}$ should be converted back to feature map representation $X_{out}^{FM} \in \mathbb{R}^{n \times h \times w}$, which is unnatural. Also, the large d makes the transformer blocks computational expensive.

3.3 FEATER

The purpose of applying transformer is to model the global correlations between a sequence of feature maps which corresponding to different human body parts. We want to preserve the inherent structure of 2D feature map representation when modeling self-attention in transformer blocks. However, as mentioned in the above section, the vanilla transformer is not able to model the self-attention given a sequence of feature maps input $X_{in}^{FM} \in \mathbb{R}^{n \times h \times w}$. All feature maps have to be flattened into $X_{in} \in \mathbb{R}^{n \times d}$ before the transformer blocks. The output flattened feature maps also need to be converted back to form the feature map representation $X_{out}^{FM} \in \mathbb{R}^{n \times h \times w}$.

Is there a better transformer architecture to deal with feature map inputs and return the feature map outputs directly and effectively? Motivated by this question, we propose a new Feature map-based transformER (FeatER) architecture which preserves the feature map representation when modeling self-attention for HPE and HMR tasks.

FeatER can be treated as the decomposition along h and w dimension of the vanilla transformer, which is illustrated in Fig. 3 (b). For w dimension stream MSA (w-MSA), the input $X_{in}^{w} \in \mathbb{R}^{n \times h \times w}$

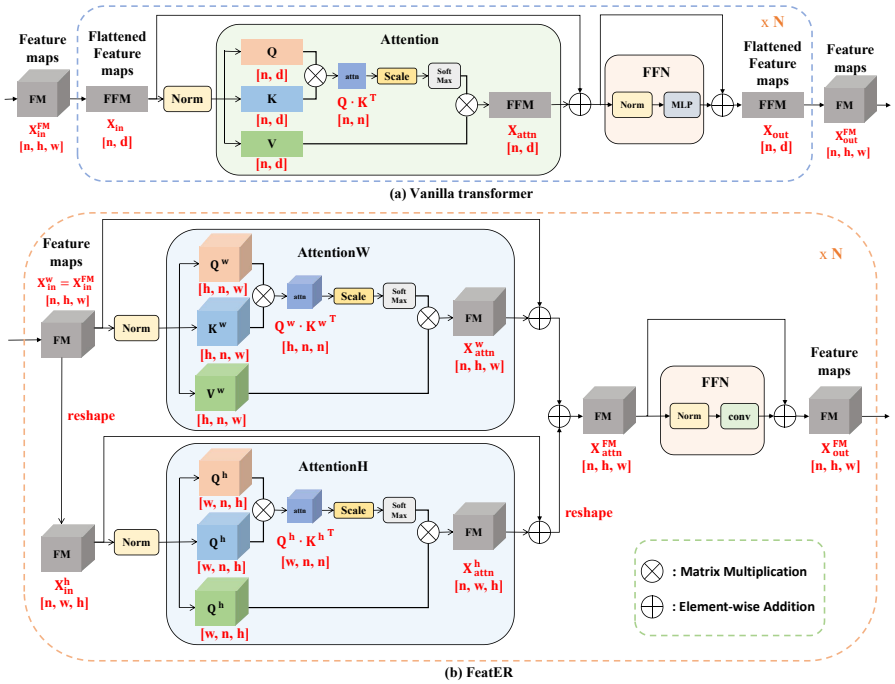

Figure 3: (a) The vanilla transformer blocks to process feature maps. (b) Our proposed FeatER blocks to process feature maps.

(equals to $X_{in}^{FM} \in \mathbb{R}^{n\times h\times w}$) is first mapped to three matrices: query matrix Q^w, key matrix K^w and value matrix V^w by three linear transformation:

$$Q^w = X_{in}^w W_{Q^w}, \quad K^w = X_{in}^w W_{K^w}, \quad V^w = X_{in}^w W_{V^w}. \tag{5}$$

where W_{Q^w}, W_{K^w} and $W_{V^w} \in \mathbb{R}^{w\times w}$.

The scaled dot product attention can be described as the following mapping function:

$$\text{AttentionW}(Q^w, K^w, V^w) = \text{Softmax}(Q^w {K^w}^\top / \sqrt{w}) V^w. \tag{6}$$

For h dimension stream MSA (h-MSA), the input $X^{FM} \in \mathbb{R}^{n\times h\times w}$ is reshape to $X_{in}^h \in \mathbb{R}^{n\times w\times h}$, then mapped to three matrices: query matrix Q^h, key matrix K^h and value matrix V^h by three linear transformation:

$$Q^h = X_{in}^h W_{Q^h}, \quad K^w = X_{in}^h W_{K^h}, \quad V^w = X_{in}^h W_{V^h}. \tag{7}$$

where W_{Q^h}, W_{K^h} and $W_{V^h} \in \mathbb{R}^{h\times h}$.

The scaled dot product attention can be described as the following mapping function:

$$\text{AttentionH}(Q^h, K^h, V^h) = \text{Softmax}(Q^h {K^h}^\top / \sqrt{h}) V^h. \tag{8}$$

Then, the FeatER block consisting of w-MSA (multi-head AttentionW), h-MSA (multi-head AttentionH) and FFN with a layer normalization operator is shown in Fig. 3 (b). The block output $X_{out}^{FM} \in \mathbb{R}^{n\times h\times w}$ keeps the same size as the input $X_{in}^{FM} \in \mathbb{R}^{n\times h\times w}$, and is represented as follows:

$$X_{attn}^{FM} = \text{w-MSA}(Q^w, K^w, V^w) + \text{h-MSA}(Q^h, K^h, V^h)^* + X_{in}^{FM} \tag{9}$$

$$X_{out}^{FM} = \text{FFN}(X_{attn}^{FM}) + X_{attn}^{FM} \tag{10}$$

where * means to reshape the matrix to the proper shape (i.e. from $\mathbb{R}^{n\times w\times h}$ to $\mathbb{R}^{n\times h\times w}$). The $\text{FFN}(\cdot)$ denotes the feed-forward network to process feature map-size input (details in Appendix B).

Complexity: A further benefit of our FeatER design is that it inherently reduces the operational computation. The theoretical computational complexity Ω of one vanilla transformer block and one

FeatER block can be approximately estimated as:

$$\Omega(\text{vanilla transformer}) = 8nd^2 + 2n^2d \tag{11}$$

$$\Omega(\text{FeatER}) = 3nhw(w+h) + 9n^2hw \tag{12}$$

If $d = h \times w$ and $h = w$, the computational complexity of FeatER can be rewritten as $\Omega(\text{FeatER}) = 6nd(\sqrt{d}) + 9n^2d$. Normally d is much larger than n, which means that the first term consumes the majority of the computational resource. The detailed computation comparison between the vanilla transformer block and the FeatER block is provided in Appendix B . Thus, FeatER reduces the computational complexity from $\mathcal{O}(d^2)$ to $\mathcal{O}(d^{3/2})$.

3.4 Feature Map Reconstruction Module

Compared with estimating 2D human pose, recovering 3D pose and mesh are more challenging due to depth ambiguity and occlusion. Some joints may be occluded by the human body or other objects in the image. In order to improve the generalization ability of our network, we apply the masking and reconstruction strategy to make our predicted human mesh more robust. Given a stack of refined feature maps $X^{FM} \in \mathbb{R}^{n \times h \times w}$, we randomly mask out m feature maps from n feature maps (the masking ratio is m/n) and utilize FeatER blocks to reconstruct feature maps. The reconstruction loss computes the mean squared error (MSE) between the reconstructed and original stack of feature maps. Then, the reconstructed stack of feature maps are used for recovering 3D pose and mesh. For the inference, the feature map masking procedure is not applied. Here we only apply the feature map reconstruction module for the 3D part. More detailed discussion is provided in Section 4.4.

4 Experiments

4.1 Implementation Details

We implemented our method FeatER for HPE and HMR tasks with Pytorch Paszke et al. (2017) using four NVIDIA RTX A5000 GPUs. We first train FeatER for 2D HPE task. The employed CNN backbone is a portion of HRNet-w32 Sun et al. (2019) (first 3 stages) pretrained on COCO Lin et al. (2014) dataset. There are 8 FeatER blocks for modeling global dependencies across all feature maps. The 2D pose head is a convolution block to output feature maps of all joints. Then, we load those pretrained weights to further train the entire pipeline as illustrated in Fig. 2 for 3D HPE and HMR tasks. There are another 8 FeatER blocks to recover feature maps in the feature map reconstruction module, where the masking ratio is 0.3. Then, we apply a 2D-3D lifting module to lift the 2D feature maps to 3D feature maps, and predict the parameters needed for the regressor. More details are provided in Appendix E and F. Next, we adopt the same regressor used in HybrIK Li et al. (2021a) to output final 3D pose and mesh. We use Adam Kingma & Ba (2014) optimizer with a learning rate of 2×10^{-4}. The batch size is 24 for each GPU.

Table 1: 2D Human Pose Estimation performance comparison with SOTA methods on the COCO validation set. The reported Params and MACs of FeatER are computed from the entire pipeline.

Model	Year	Input size	Params (M)	MACs (G)	AP ↑	AP50 ↑	AP75 ↑	AP(M) ↑	AP(L) ↑	AR ↑
Compared with Small Networks										
DY-MobileNetV2 Chen et al. (2020a)	CVPR 2020	256×192	16.1	1.0	68.2	88.4	76.0	65.0	74.7	74.2
Transpose_H_S Yang et al. (2021)	ICCV 2021	256×192	8.0	10.2	74.2	-	-	-	-	78.0
Tokenpose_B Li et al. (2021e)	ICCV 2021	256×192	13.5	5.7	74.7	89.8	81.4	**71.3**	81.4	**80.0**
HRFormer_S Yuan et al. (2021)	NeurIPS 2021	256×192	7.8	2.8	74.0	**90.2**	81.2	70.4	80.7	79.4
FeatER		256×192	8.1	5.4	**74.9**	89.8	**81.6**	71.2	**81.7**	**80.0**
Compared with Large Networks										
SimpleBaseline Xiao et al. (2018)	ECCV 2018	256×192	34.0	8.9	70.4	88.6	78.3	-	-	76.3
HRNet_W32 Sun et al. (2019)	CVPR 2019	256×192	28.5	7.1	74.4	**90.5**	**81.9**	-	-	78.9
PRTR Li et al. (2021b)	CVPR 2021	384×288	57.2	21.6	73.1	89.4	79.8	68.8	80.4	79.8
PRTR Li et al. (2021b)	CVPR 2021	512×384	57.2	37.8	73.3	89.2	79.9	69.0	80.9	**80.2**
FeatER		256×192	8.1	5.4	**74.9**	89.8	81.6	**71.2**	**81.7**	80.0

4.2 2D HPE

Dataset and evaluation metrics: We conduct the 2D HPE experiment on the COCO Lin et al. (2014) dataset, which contains over 200,000 images and 250,000 person instances. There are 17 keypoints labeled for each person instance. We train our model on the COCO train2017 set and evaluate on the COCOval2017 set, with the experiment setting following Li et al. (2021e). The evaluation metrics we adopt are standard Average Precision (AP) and Average Recall (AR) Chen et al. (2020b).

Results: Table 1 compares FeatER with previous SOTA methods for 2D HPE on COCO validation set including the total parameters (Params) and Multiply–Accumulate operations (MACs). Since FeatER is a lightweight model, we first compare with previous lightweight methods (Params < 20M and MACs < 20G) with the input image size of 256×192. Our FeatER achieves the best results on 4 (AP, AP75, AP(L), and AR) of the 6 evaluation metrics. When compared with the SOTA lightweight transformer-based method Tokenpose_B Li et al. (2021e), FeatER only requires 60% of Params and 95% of MACs while improving 0.2 AP, 0.2 AP75, and 0.3 AP(L).

As an efficient lightweight model, FeatER can even achieve competitive performance with methods of large models while showing a significant reduction in Params and MACs. For instance, FeatER outperforms PRTR Sun et al. (2019) in terms of AP, AP50, AP75, AP(M) and AP(L) with only 14% of Params and 14% of MACs.

4.3 3D HPE AND HMR

Datasets and evaluation metrics: We evaluate FeatER for 3D HPE and HMR on Human3.6M Ionescu et al. (2014) and 3DPW von Marcard et al. (2018) datasets. Human3.6M is one of the largest indoor datasets which contains 3.6M video frames in 17 actions performed by 11 actors. Following previous work Kolotouros et al. (2019); Choi et al. (2020); Lin et al. (2021b), we use 5 subjects (S1, S5, S6, S7, S8) for training and 2 subjects (S9, S11) for testing. 3DPW is an in-the-wild dataset that is composed of 60 video sequences (51K frames). The accurate 3D mesh annotations are provided. We follow the standard train/test split from the dataset. Mean Per Joint Position Error (MPJPE) Chen et al. (2020b) and the MPJPE after Procrustes alignment (PA-MPJPE) Chen et al. (2020b) are reported on Human3.6M. Besides these, to evaluate the reconstructed mesh, the Mean Per Vertex Error (MPVE) is reported on 3DPW. Following Kolotouros et al. (2019); Li et al. (2021a); Moon & Lee (2020), Human3.6M, MPI-INF-3DHP, COCO, 3DPW are used for mixed training (we indicate the result with or without 3DPW training set in Table 2 for fair comparisons). Following previous work Kolotouros et al. (2019)Lin et al. (2021b)Lin et al. (2021a), the 3D human pose is calculated from the estimated mesh multiplied with the defined joint regression matrix.

Results: Table 2 compares FeatER with previous SOTA methods for 3D HPE and HMR on Human3.6M and 3DPW including the Params and MACs. FeatER outperforms the SOTA methods on Human3.6M and 3DPW datasets with very low Params and MACs. For Human3.6M dataset, FeatER reduces 1.3 of MPJPE and 1.7 of PA-MPJPE compared with SOTA method MeshGraphormer Lin et al. (2021a).

For 3DPW, FeatER improves the MPJPE from 89.7 Khirodkar et al. (2022) to 88.4, the PA-MPJPE from 55.8 Choi et al. (2021) to 54.5, and MPVE from 107.1 Khirodkar et al. (2022) to 105.6 without using 3DPW training set. When using 3DPW training set during training, FeatER also shows superior performance compared to MeshGraphormer Lin et al. (2021a). Moreover, FeatER reduces the memory and computational costs significantly (only 5% of Params and 16% of MACs comapred with MeshGraphormer Lin et al. (2021a)). Thus, FeatER is a much more time and resource efficient model for HPE and HMR tasks with exceptional performance.

Table 2: 3D Pose and Mesh performance comparison with SOTA methods on Human3.6M and 3DPW datasets. * indicates 3DPW training set is used during training.

				Human3.6M		3DPW		
Model	Year	Params (M)	MACs (G)	MPJPE↓	PA-MPJPE↓	MPJPE↓	PA-MPJPE↓	MPVE↓
SPIN Kolotouros et al. (2019)	ICCV 2019	-	-	62.5	41.1	96.9	59.2	116.4
VIBE Kocabas et al. (2020)	CVPR 2020	-	-	65.6	41.4	93.5	56.5	113.4
I2LMeshNet Moon & Lee (2020)	ECCV 2020	140.5	36.6	55.7	41.1	93.2	57.7	-
TCMR Choi et al. (2021)	CVPR 2021	-	-	62.3	41.1	95.0	55.8	111.5
ProHMR Kolotouros et al. (2021)	ICCV 2021	-	-	-	41.2	-	59.8	-
PyMAF Zhang et al. (2021)	ICCV 2021	45.2	10.6	57.7	40.5	92.8	58.9	110.1
OCHMR Khirodkar et al. (2022)	CVPR 2022	-	-	-	-	89.7	58.3	107.1
FeatER		11.4	8.8	**49.9**	**32.8**	**88.4**	**54.5**	**105.6**
VIBE* Kocabas et al. (2020)	CVPR 2020	-	-	65.6	41.4	82.9	51.9	99.1
DSR* Dwivedi et al. (2021)	ICCV 2021	-	-	60.9	40.3	85.7	51.7	99.5
HybrIK(ResNet34)* Li et al. (2021a)	CVPR 2021	27.6	12.7	57.3	36.2	75.3	**45.2**	87.9
TCFormer* Zeng et al. (2022)	CVPR 2022	-	-	62.9	42.8	80.6	49.3	-
METRO* Lin et al. (2021b)	CVPR 2021	229.2	56.6	54.0	36.7	77.1	47.9	88.2
FastMETRO* Cho et al. (2022)	ECCV 2022	48.5	28.3	53.9	37.3	77.9	48.3	90.6
MeshGraphormer* Lin et al. (2021a)	ICCV 2021	226.5	56.6	51.2	34.5	74.7	45.6	87.7
FeatER*		11.4	8.8	**49.9**	**32.8**	**73.4**	45.9	**86.9**

4.4 ABLATION STUDY

We conduct the ablation study on COCO, Human3.6M, and 3DPW datasets. The train/test split, experiments setting, and evaluation metrics are the same as in Sections 4.2 and 4.3.

Effectiveness of FeatER: We compare the vanilla transformer architecture in Fig. 3 (a) with our FeatER block in Fig. 3 (b) on 2D HPE, 3D HPE, and HMR tasks in Table 3 and 4, respectively.

'No transformer' means we do not employ transformer to refine the coarse feature maps extracted by the CNN backbone. The 'VanillaTransformer' indicates that we utilize the vanilla transformer as described in Section 3.2 instead of the proposed FeatER blocks to refine the coarse feature maps in the pipeline. For fair comparisons, given the input of the blocks $X_{in}^{FM} \in \mathbb{R}^{n\times h\times w}$, FeatER blocks return the output $X_{out}^{FM} \in \mathbb{R}^{n\times h\times w}$. 'VanillaTransformer' first flattens the input to $X_{in} \in \mathbb{R}^{n\times d}$ and returns $X_{out} \in \mathbb{R}^{n\times d}$. Next, the flattened output is reshaped to $X_{out}^{FM} \in \mathbb{R}^{n\times h\times w}$ following the feature map format. 'VanillaTransformer_S' is the small version of vanilla transformer, which has similar computational complexity with FeatER blocks and the embedding dimension is shrank to $d = 384$. 'VanillaTransformer_L' is the large version of vanilla transformer, which requires more memory and computational costs.

In Table 3, without employing transformer, the network requires less Params and MACs but the performance is worse than others. Once transformer is applied, FeatER outperforms VanillaTransformer_S by a large margin with similar MACs and 42% of Params. Even compared with VanillaTransformer_L, FeatER can achieve competitive results while only requires 12% Params and 55% MACs.

Table 3: Ablation study on transformer design for 2D HPE task on COCO validation set.

Model	Input size	Params (M)	MACs (G)	AP ↑	AP50 ↑	AP75 ↑	AP(M) ↑	AP(L) ↑	AR ↑
No transformer	256×192	7.2	4.4	72.9	87.8	79.1	69.0	78.3	76.4
VanillaTransformer_S	256×192	19.5	5.4	74.0	89.2	80.4	70.4	79.5	78.4
VanillaTransformer_L	256×192	69.1	9.8	75.1	90.2	81.3	71.0	81.8	80.0
FeatER	256×192	8.1	5.4	74.9	89.8	81.6	71.2	81.7	80.0

Table 4: Ablation study on transformer design for 3D HPE and HMR tasks on Human3.6M and 3DPW datasets.

			Human3.6M		3DPW		
Model	Params (M)	MACs (G)	MPJPE ↓	PA-MPJPE ↓	MPJPE ↓	PA-MPJPE ↓	MPVE ↓
No transformer	9.7	6.9	54.4	39.1	92.5	56.9	109.6
VanillaTransformer_S	30.5	8.9	52.7	37.6	90.6	55.3	107.4
VanillaTransformer_L	127.7	18.2	48.3	33.7	88.3	54.8	105.6
FeatER	11.4	8.8	49.9	32.8	88.4	54.5	105.6

In Table 4, we observe a similar trend where FeatER surpasses VanillaTransformer_S by a large margin with similar MACs and 37% of Params. While only requiring 9% Params and 48% MACs, FeatER achieves comparable results compared with using VanillaTransformer_L.

We can conclude that FeatER is an extremely efficient network with a strong modeling capability, which is more suitable for 2D HPE, 3D HPE, and HMR tasks. More analysis and feature maps visualization are shown in Appendix C.

Effectiveness of using feature map reconstruction module: The purpose of applying the feature map reconstruction module is to improve the generalization ability of our network. In our current design, the Reconstruction Module is for 3D pose and mesh tasks. The occlusion and depth ambiguity make 3D HPE and HMR more challenging than 2D HPE. Thus 3D HPE and HMR can be more benefited by adding the Reconstruction Module. As shown in Table 5, once the Reconstruction Module is added, the performance can be improved. If we move the Reconstruction Module for 2D HPE, although the performance of 2D HPE can be increased slightly, the performance of 3D HPE and HMR can not be boosted significantly. If we use two Reconstruction Modules, one for 2D HPE and another for 3D HPE and HMR. The performance also can not be further improved. Moreover, the Params and MACs are increased, which is not what we want. Thus, putting the Feature Map Reconstruction Module in the 3D part is the optimal solution to trade-off accuracy and efficiency. More analysis and results about the feature map reconstruction module are provided in Appendix D.

Table 5: Ablation study on the different positions of the Feature Map Reconstruction Module.

			COCO		**Human3.6M**	**3DPW**
	Params (M)	MACs (G)	AP ↑	AR ↑	MPJPE ↓	MPVE ↓
No Reconstruction Module	10.4	7.7	74.9	80.0	53.3	94.5
In the 2D Part	11.4	8.8	75.3	80.2	52.8	88.7
In the 3D Part (FeatER's design)	11.4	8.8	74.9	80.0	49.9	86.9
In both the 2D part and 3D part	12.5	10.0	75.3	80.2	49.8	87.1

4.5 QUALITATIVE RESULTS

To show the qualitative results of the proposed FeatER for human reconstruction (2D HPE, 3D HPE, and HMR), we use the challenging COCO dataset which consists of in-the-wild images. Given various input images, FeatER can estimate reliable human poses and meshes as shown in Fig. 4. When comparing with I2LMeshNet Moon & Lee (2020) and PyMAF Zhang et al. (2021) in Fig. 5, the areas highlighted by red circles indicate that FeatER outputs more accurate meshes under challenging scenarios. We provide more visual examples on different datasets in Appendix G.

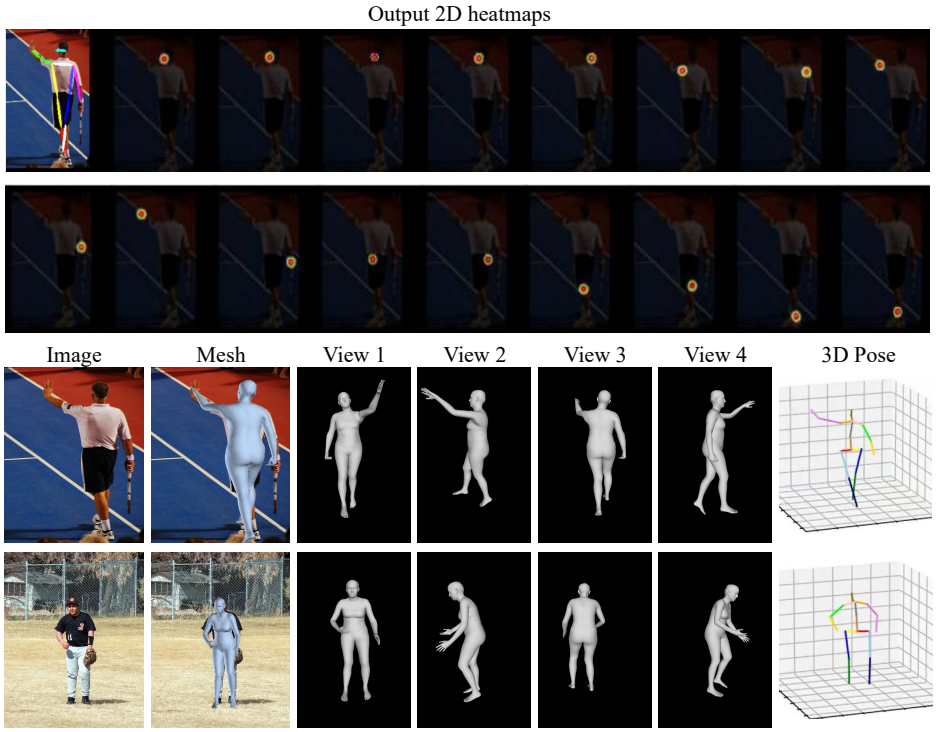

Figure 4: Qualitative results of the proposed FeatER. Images are taken from the in-the-wild COCO Lin et al. (2014) dataset.

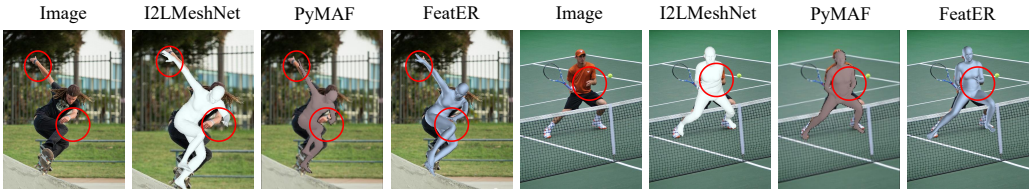

Figure 5: Qualitative comparison with other methods. Images are taken from the in-the-wild COCO Lin et al. (2014) dataset. The red circles highlight locations where FeatER is more accurate than others. We follow previous work Lin et al. (2021b;a); Moon & Lee (2020); Zhang et al. (2021) to visualize human mesh using the SMPL *gender neutral* model.

5 CONCLUSION

In this paper, we present FeatER, a novel feature map-based transformer architecture for HPE and HMR. FeatER can preserve the feature map representations and effectively model global correlations between them via self-attention. By performing decomposition with th w and h dimensions, FeatER significantly reduces the computational complexity compared with vanilla transformer architecture. Furthermore, the introduced feature map reconstruction module improves the robustness of the estimated human pose and mesh. Extensive experiments show that FeatER improves the performance while significantly reducing the computational cost for HPE and HMR tasks.

REFERENCES

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

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11030–11039, 2020a.

Yucheng Chen, Yingli Tian, and Mingyi He. Monocular human pose estimation: A survey of deep learning-based methods. *Computer Vision and Image Understanding*, 192:102897, 2020b.

Junhyeong Cho, Kim Youwang, and Tae-Hyun Oh. Cross-attention of disentangled modalities for 3d human mesh recovery with transformers. In *European Conference on Computer Vision (ECCV)*, 2022.

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.

Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1964–1973, June 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Sai Kumar Dwivedi, Nikos Athanasiou, Muhammed Kocabas, and Michael J Black. Learning to regress bodies from images using differentiable semantic rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11250–11259, 2021.

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*, pp. 770–778, 2016.

Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15013–15022, October 2021.

Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, jul 2014.

Rawal Khirodkar, Shashank Tripathi, and Kris Kitani. Occluded human mesh recovery. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020.

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*, pp. 2252–2261, 2019.

Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11605–11614, 2021.

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*, pp. 3383–3393, 2021a.

Ke Li, Shijie Wang, Xiang Zhang, Yifan Xu, Weijian Xu, and Zhuowen Tu. Pose recognition with cascade transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1944–1953, 2021b.

Ming Li, Jun Liu, Ce Zheng, Xinming Huang, and Ziming Zhang. Exploiting multi-view part-wise correlation via an efficient transformer for vehicle re-identification. *IEEE Transactions on Multimedia*, 2021c.

Wenhao Li, Hong Liu, Hao Tang, Pichao Wang, and Luc Van Gool. Mhformer: Multi-hypothesis transformer for 3d human pose estimation. *arXiv preprint arXiv:2111.12707*, 2021d.

Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, Pichao Wang, and Wenming Yang. Exploiting temporal contexts with strided transformer for 3d human pose estimation. *IEEE Transactions on Multimedia*, 2022.

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*, pp. 11313–11322, 2021e.

Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021a.

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*, pp. 1954–1963, 2021b.

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*, pp. 740–755. Springer, 2014.

Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2949–2958, 2021.

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

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2906–2917, 2021.

Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, 2020.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

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*, pp. 5693–5703, 2019.

Yating Tian, Hongwen Zhang, Yebin Liu, and Limin Wang. Recovering 3d human mesh from monocular images: A survey. *arXiv preprint arXiv:2203.01923*, 2022.

Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *European Conference on Computer Vision (ECCV)*, sep 2018.

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

Fanglei Xue, Qiangchang Wang, and Guodong Guo. Transfer: Learning relation-aware facial expression representations with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3601–3610, 2021.

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*, pp. 11802–11812, 2021.

Changqian Yu, Bin Xiao, Changxin Gao, Lu Yuan, Lei Zhang, Nong Sang, and Jingdong Wang. Lite-hrnet: A lightweight high-resolution network. In *CVPR*, 2021.

Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution transformer for dense prediction. *arXiv preprint arXiv:2110.09408*, 2021.

Mihai Zanfir, Andrei Zanfir, Eduard Gabriel Bazavan, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Thundr: Transformer-based 3d human reconstruction with markers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12971–12980, 2021.

Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11101–11111, 2022.

Feng Zhang, Xiatian Zhu, Hanbin Dai, Mao Ye, and Ce Zhu. Distribution-aware coordinate representation for human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. Pymaf: 3d human pose and shape regression with pyramidal mesh alignment feedback loop. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6848–6856, 2018.

Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11656–11665, October 2021.

Ce Zheng, Matias Mendieta, Pu Wang, Aidong Lu, and Chen Chen. A lightweight graph transformer network for human mesh reconstruction from 2d human pose. *ACM multimedia*, 2022.

APPENDIX

A SOCIETAL IMPACT

While our network does not raise a direct negative societal impact, it may be used by some applications for malicious purposes, such as unwarranted surveillance. To avoid possible negative societal impact, we urge the readers to limit our network to ethical and legal use-cases.

B DETAILS OF COMPLEXITY COMPARISON

In Table 6, 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 are 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.

Table 6: 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	$a^w_1 = Q^w K^{wT}$	$Q^w[h, n, w]$,$K^{wT}[h, w, n]$	$a^w_1\,[h, n, n]$	torch.matmul	n^2hw
$x_{attn} = a_1V$	$a_1[n, n]$, $V[n, d]$	$x_{attn}[n, d]$	torch.matmul	n^2d	$x^w_{attn} = a^w_1 V^w$	$a^w_1[h, n, n]$, $V[h, n, w]$	$x^w_{attn}[h, n, w]$	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)	$3nh^2w$
					$a^h_1 = Q^h K^{hT}$	$Q^h[w, n, h]$,$K^{hT}[w, h, n]$	$a^h_1\,[w, n, n]$	torch.matmul	n^2hw
					$x^h_{attn} = a^h_1 V^h$	$a^h_1\,[w, n, n]$,$V^h\,[w, n, h]$	$x^w_{attn}[w, n, h]$	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 $w = h$				

C 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. 6. These examples demonstrate that our proposed feature map-based transformer (FeatER) blocks can successfully refine the coarse feature maps by hinting more accurate joint locations, thereby improving the performance of human reconstructions tasks (2D HPE, 3D HPE, and HMR).

D 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 7. The performance is improved in all cases, including for the most challenging actions on the Human3.6M indoor dataset with heavy occlusion such as Photo, SitD (sitting down), and WalkD

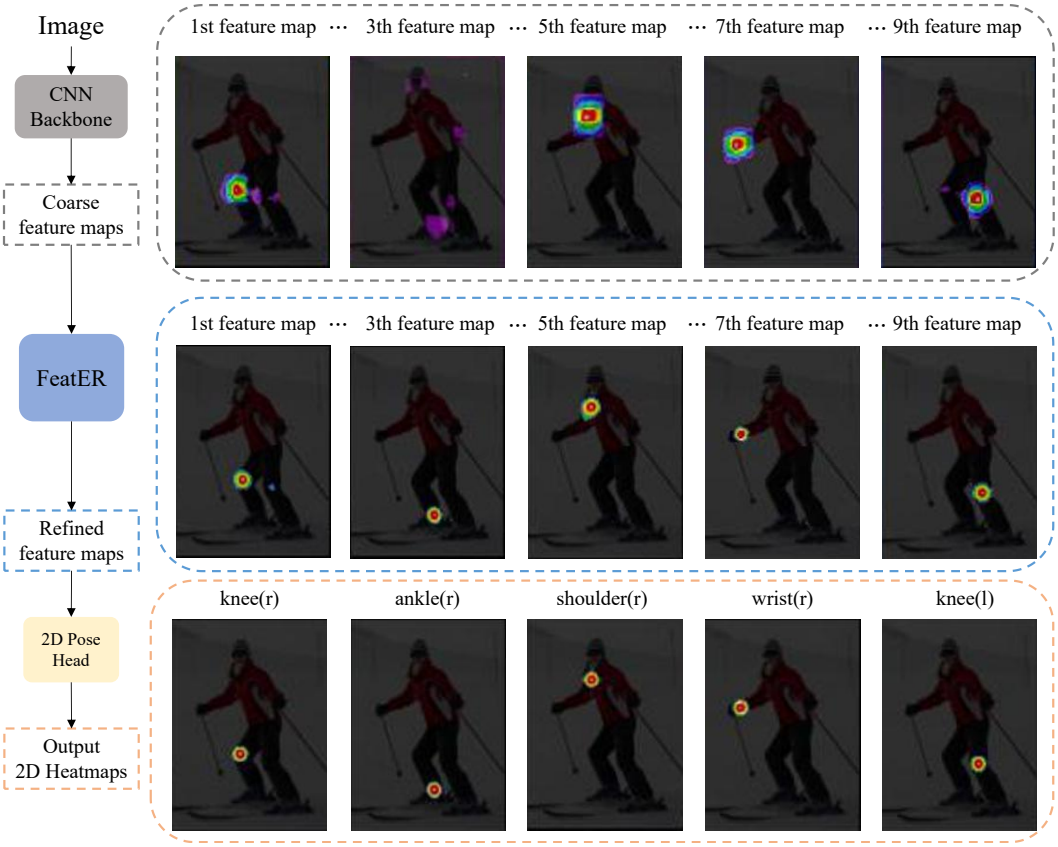

Figure 6: Visualization of coarse feature maps (extracted by CNN backbone) and refined feature maps (refined by FeatER).

(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 been decreased. Therefore, through this analysis, we validate the effectiveness of using the feature map reconstruction module.

Table 7: 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

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. 7. We set the masking ratio to be 0.3 since it provides the best results on both Human3.6M and 3DPW datasets.

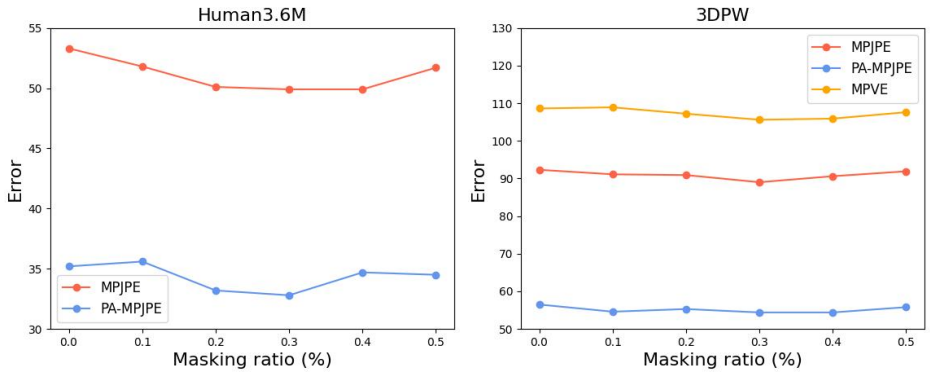

Figure 7: Evaluation of different masking ratios in the feature map reconstruction module.

E 2D-3D Lifting Module

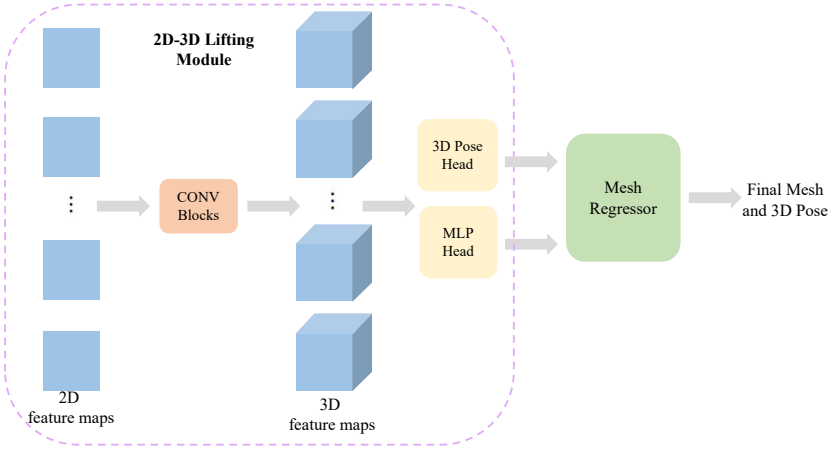

Figure 8: The architecture of the 2D-3D Lifting Module

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 intermidate 3D Pose can be obtained by a 3D pose head. The MLP head outputs the parameters for the mesh regressor. The architecture of the 2D-3D Lifting Module is shown in Fig. 8.

F LOSS FUNCTION

2D HPE

We first train our FeatER on COCO dataset for the 2D HPE task. Following Sun et al. (2019); Li et al. (2021e), we apply the Mean 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 2D pose is defined as follows:

$$\mathcal{L}_{2D-Pose} = \|HM - HM^{GT}\|^2 \tag{13}$$

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 Choi et al. (2020); Lin et al. (2021b); Li et al. (2021a). K is the number of joints.

$$\mathcal{L}_{3D-Pose} = \frac{1}{K}\sum_{i=1}^{K} \|J_i - J_i^{GT}\|_1 \tag{14}$$

Following Li et al. (2021a), we use the SMPL Loper et al. (2015) 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 Loper et al. (2015). The reconstruction loss is the Mean Square Error (MSE) between the target feature maps and reconstructed feature maps. The overall loss is defined as follows:

$$\mathcal{L}_{overall} = \mathcal{L}_{3D-Pose} + w_1\|\beta - \beta^{GT}\| + w_2\|\theta - \theta^{GT}\| + w_3\mathcal{L}_{reconstruction} \tag{15}$$

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

G MORE QUALITATIVE RESULTS

2D Heatmap and Human Mesh Reconstruction (HMR) Visualization

Fig. 9 provides visulization of 17 heatmaps (COCO Lin et al. (2014) 17 joints format) and the predicted 2D poses of the input images. The visualization of Human3.6M and 3DPW dataset are shown in Figs. 11. Figs. 10 and 12 show the HMR visualization of FeatER on several in-the-wild images from the COCO Lin et al. (2014) dataset. FeatER can estimate accurate human meshes of the given images with regular human articulation in Fig. 10. For some very challenging cases as shown in Fig. 12, FeatER can still output reliable human meshes.

When comparing with the state-of-the-art HMR method METRO Lin et al. (2021b), FeatER clearly outperforms METRO with only 5% of Params and 16% of MACs on these in-the-wild images (taken from the COCO Lin et al. (2014) dataset) as depicted in Fig. 13, 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. 10 and 12, there are still some inaccurate and failure cases. As presented in Fig. 14 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. 14 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.

Figure 9: 2D heatmaps visualization of the proposed FeatER. Images are taken from the COCO validation set Lin et al. (2014).

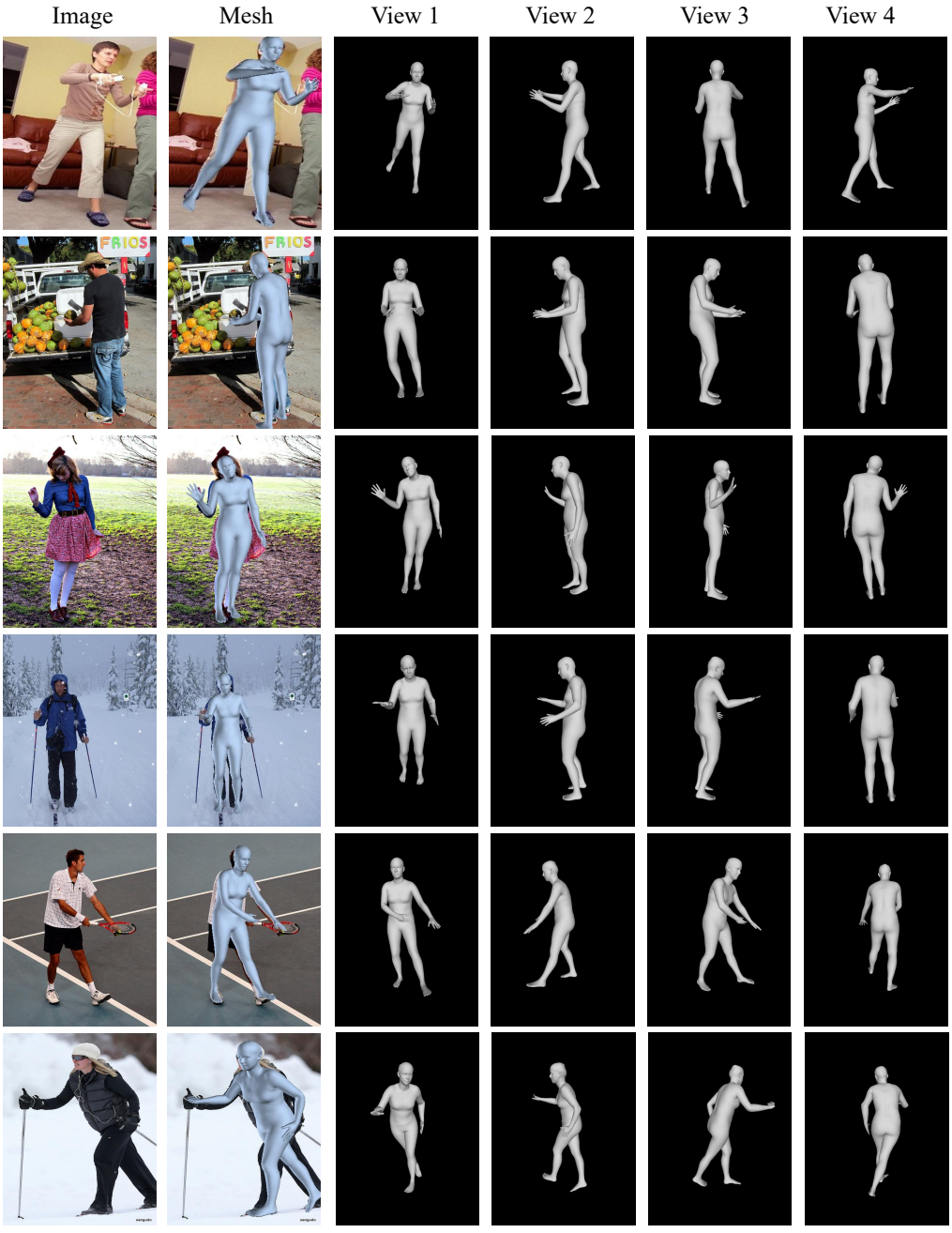

Figure 10: Mesh reconstruction qualitative results of the proposed FeatER. Images are taken from the in-the-wild COCO Lin et al. (2014) dataset.

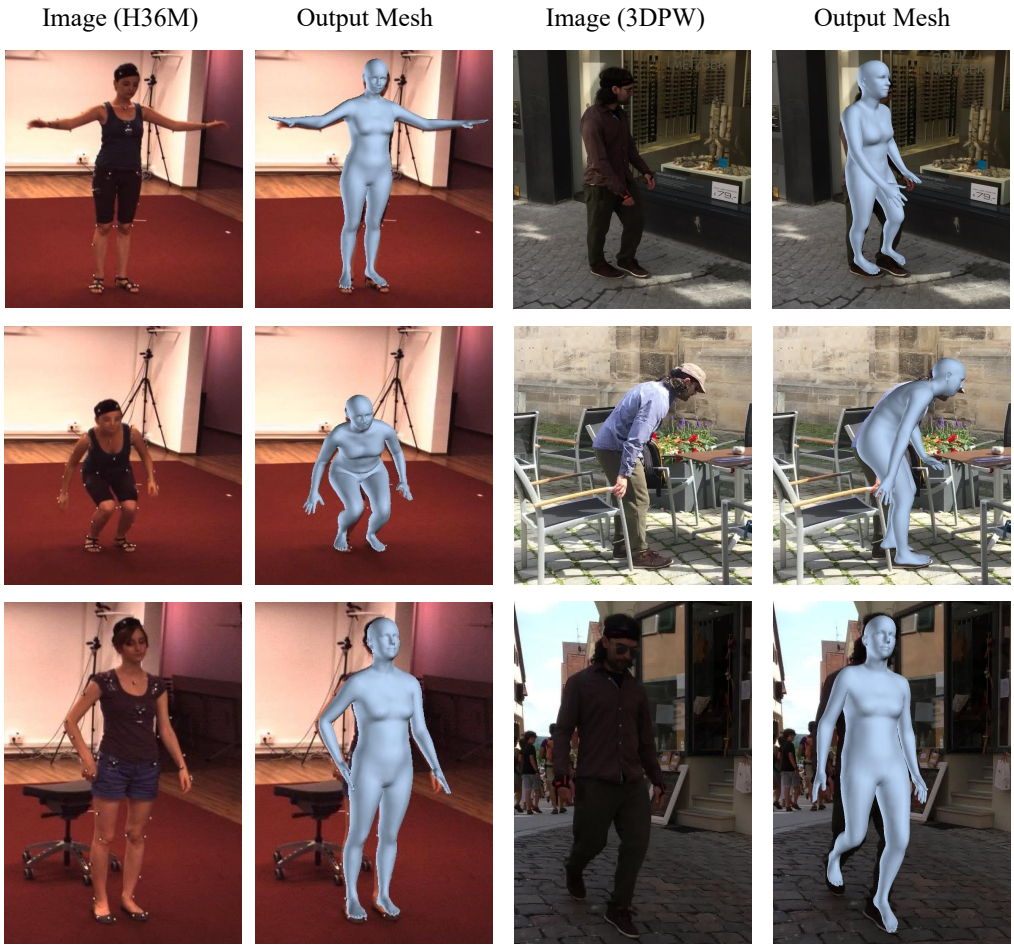

Figure 11: Mesh reconstruction qualitative results of the proposed FeatER. Images are taken from the Human3.6M dataset and 3DPW dataset.

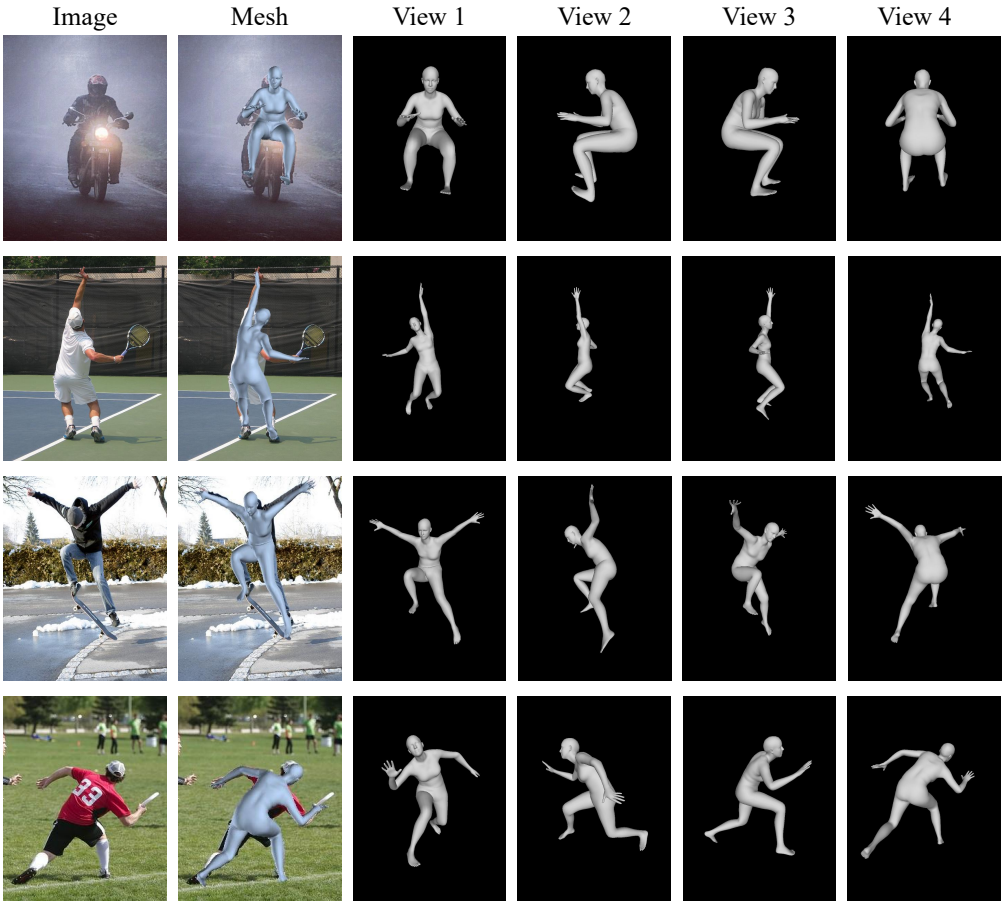

Figure 12: Mesh reconstruction qualitative results of the proposed FeatER for more challenging cases. Images are taken from the in-the-wild COCO Lin et al. (2014) dataset.

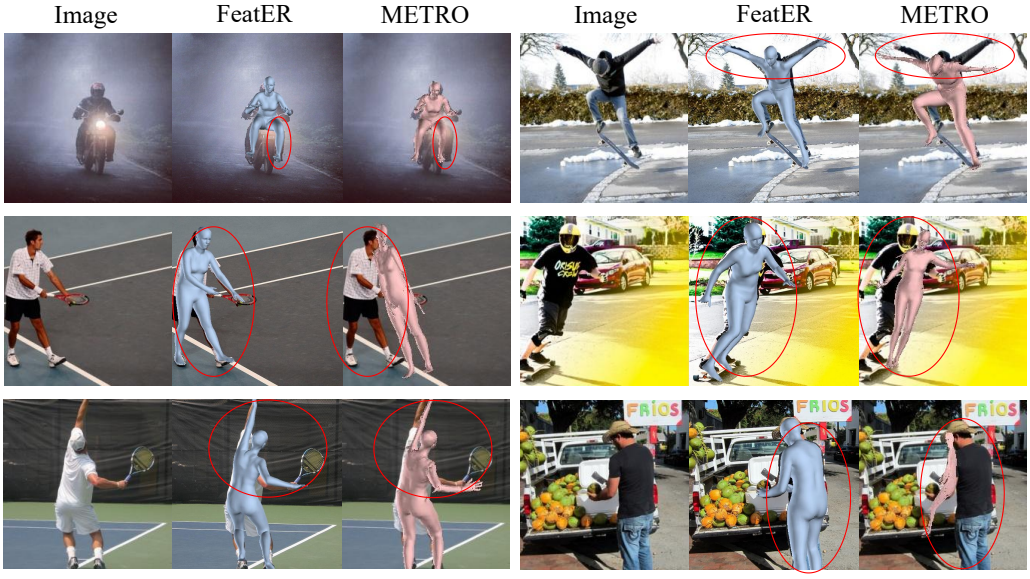

Figure 13: Qualitative comparison with the state-of-the-art HMR method METRO Lin et al. (2021b). Images are taken from the in-the-wild COCO Lin et al. (2014) dataset. The red circles highlight locations where FeatER is more accurate than METRO.

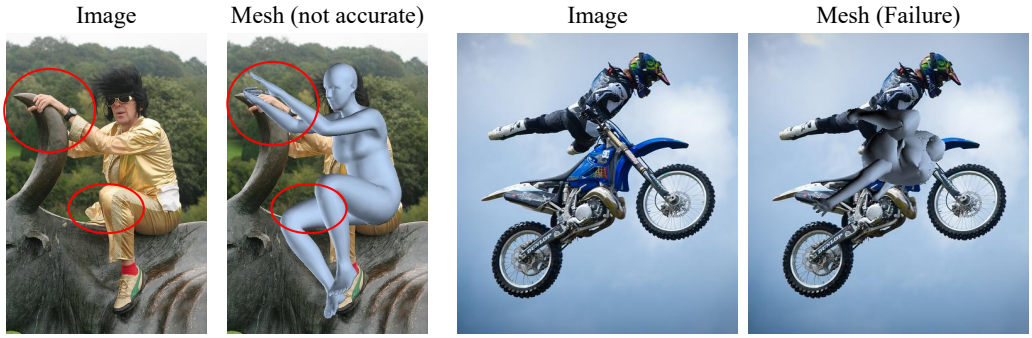

Figure 14: Left: Inaccurate estimated mesh due to heavy occlusion. Right: Failure estimated mesh due to complex human body articulation.