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

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

Abstract

Recently, vision transformers have shown great success in a set of human reconstruction tasks such as 2D/3D human pose estimation (2D/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 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 that preserves the inherent structure of feature map representations when modeling attention while reducing 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 and 16% of MACs on Human3.6M and 3DPW datasets. The project webpage is https://zczcwh.github.io/feater_page/.

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 toward 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.

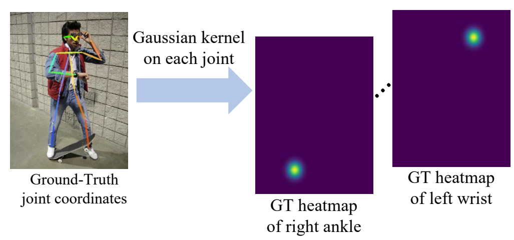

Figure 1. Generating heatmaps from joint coordinates.

With the blooming of deep learning techniques, Convolutional Neural Network (CNN) [10, 32, 33] architectures have been extensively utilized in vision tasks and have achieved impressive performance. Most existing HPE and HMR models [13, 33] utilize CNN-based architectures (such as ResNet [10] and HRNet [33]) 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 [1, 33, 43].

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 [23–25, 38, 47]. 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 [8] 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 $[n, h, w]$ need 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 caused by the flattening process makes the transformer computationally expensive. This is not suitable for real-world applications of HPE and HMR, which often demand real-time processing capabilities on deployed devices (e.g. AR/VR headsets).

Therefore, 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 are 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 [25] with only requiring 5% of Params and 16% of MACs).

2. Related work

Since Vision Transformer (ViT) [8] 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 [27, 29], facial expression recognition [37], and re-identification [11, 20]. Since the related work of HPE and HMR is vast, we refer interested readers to the recent and comprehensive surveys: [4] for HPE and [34] for HMR. In this section, we discuss the more relevant transformer-based approaches.

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 [19, 23, 38] utilize transformers in 2D HPE. TransPose [38] uses a transformer to capture the spatial relationships between keypoint joints. PRTR [19] builds cascade transformers with encoders and decoders based on DETR [2]. Although achieving impressive performance, TransPose [38] and PRTR [19] suffer from heavy computational costs. HRFormer [40] integrates transformer blocks in the HRNet structure to output 2D human pose. TokenPose [23] 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 [47] and Li et al. [21] first apply transformers in 3D HPE. MHFormer [22] generates multiple plausible pose hypotheses using transformers to lift 2D pose input to 3D pose output. As 2D-3D lifting approaches, these methods [21, 22, 47] 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 [41] introduces a model-free transformer-based architecture for human mesh reconstruction. GTRS [46] proposes a graph transformer architecture with parallel design to estimate 3D human mesh only from detected 2D human pose. METRO [24] combines a CNN backbone with a transformer network to regress human mesh vertices directly for HMR. MeshGraphormer [25] 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

[1]For example, there are 32 feature maps with overall dimension $[32, 64, 64]$. For a vanilla transformer, without discarding information, the feature maps need 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.

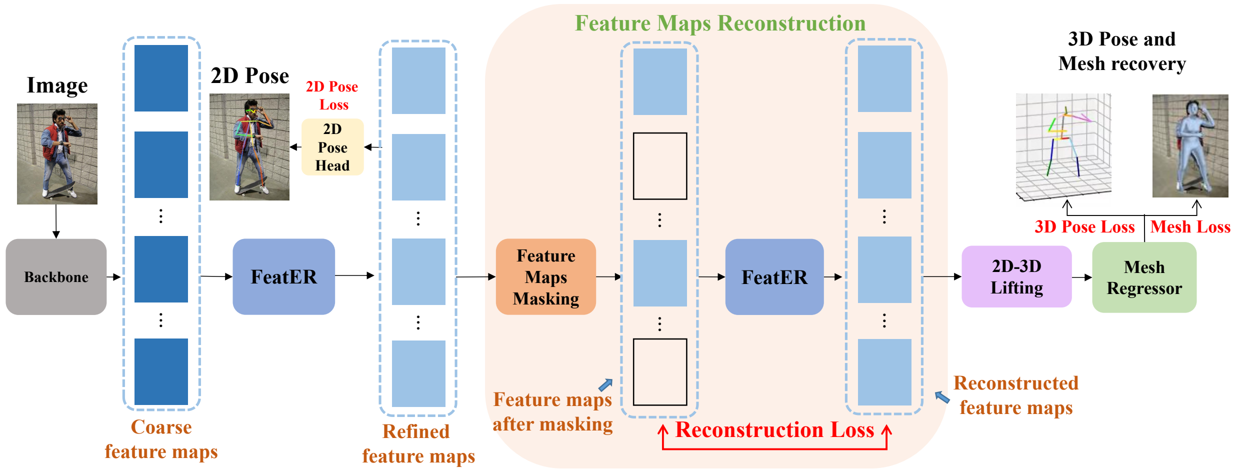

Figure 2. An overview of our proposed network for 2D HPE, 3D HPE, and HMR tasks. The coarse feature maps are extracted by the CNN backbone and refined by FeatER blocks. The 2D pose can be obtained by a 2D pose head. Then, we apply a feature map reconstruction module to improve the robustness of the predicted 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.

methods for HPE and HMR [24, 25] mainly pursue higher accuracy without considering computation and memory cost. While less studied, model efficiency is also a key characteristic of HPE and HMR applications. Lite-HRNet [39] applies the efficient shuffle block in ShuffleNet [28, 45] to HRNet [33], but it is only limited to 2D HPE. GTRS [46] 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 [9, 25]. Our FeatER is an efficient network that can outperform SOTA methods while reducing computation and memory costs significantly.

3. Methodology

3.1. Overview Architecture

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 HybrIK [18] output the estimated 3D pose and mesh.

3.2. Preliminaries of Vanilla Transformer

The input of a vanilla transformer [8] 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:

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

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:

$$X_{attn} = \mathrm{MSA}(Q, K, V) + X_{in} \tag{3}$$
$$X_{out} = \mathrm{FFN}(X_{attn}) + X_{attn} \tag{4}$$

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

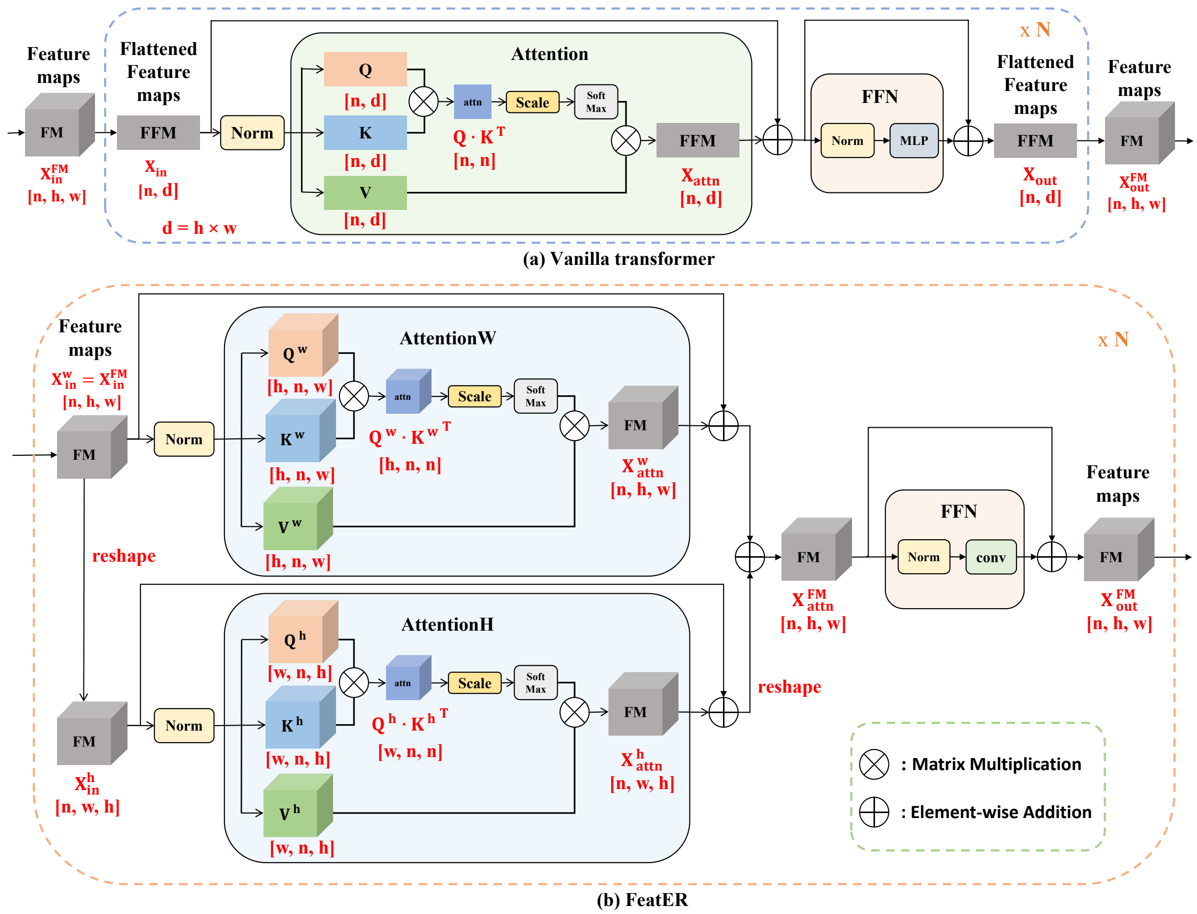

Figure 3. (a) The vanilla transformer blocks to process feature maps. (b) Our proposed FeatER blocks to process feature maps.

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 computationally expensive.

3.3. FeatER

The purpose of applying transformer is to model the global correlations between a sequence of feature maps that 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 that 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 $X_{in}^{w} \in \mathbb{R}^{n \times h \times w}$ (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:

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

$$X_{attn}^{FM} = X_{attn}^{FM} + X_{in}^{FM} \tag{10}$$

$$X_{out}^{FM} = \text{FFN}(X_{attn}^{FM}) + X_{attn}^{FM} \tag{11}$$

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 Supplementary A).

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:

$$\Omega(\text{vanilla transformer}) = 8nd^2 + 2n^2 d \tag{12}$$

$$\Omega(\text{FeatER}) = 3nhw(w+h) + 9n^2 hw \tag{13}$$

If $d = h \times w$ and $h = w$, the computational complexity of FeatER can be rewritten as $\Omega(\text{FeatER}) = 6nd(\sqrt{d}) + 9n^2 d$. 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 Supplementary A. 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 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 is 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 [31], which is trained on four NVIDIA RTX A5000 GPUs. We first train FeatER for the 2D HPE task. The employed CNN backbone is a portion of HRNet-w32 [33] (first 3 stages) pretrained on COCO [26] dataset. There are 8 FeatER blocks for modeling global dependencies across all feature maps. The 2D pose head is a convolution block to output heatmaps 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 Supplementary D and E. Next, we adopt the mesh regressor HybrIK [18] to output the final 3D pose and mesh. We use Adam [14] optimizer with a learning rate of 2×10^{-4}. The batch size is 24 for each GPU.

4.2. 2D HPE

Dataset and evaluation metrics: We conduct the 2D HPE experiment on the COCO [26] 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 [23]. The evaluation metrics we adopt are standard Average Precision (AP) and Average Recall (AR) [4].

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 $\leq$ 20M and MACs $\leq$ 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

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 [3]	CVPR 2020	256×192	16.1	1.0	68.2	88.4	76.0	65.0	74.7	74.2
HRFormer_S [40]	NeurIPS 2021	256×192	7.8	2.8	74.0	**90.2**	81.2	70.4	80.7	79.4
Transpose_H_S [38]	ICCV 2021	256×192	8.0	10.2	74.2	-	-	-	-	78.0
Tokenpose_B [23]	ICCV 2021	256×192	13.5	5.7	74.7	89.8	81.4	**71.3**	81.4	**80.0**
FeatER		256×192	8.1	5.4	**74.9**	89.8	**81.6**	71.2	**81.7**	**80.0**
Compared with Large Networks										
SimpleBaseline [36]	ECCV 2018	256×192	34.0	8.9	70.4	88.6	78.3	-	-	76.3
HRNet_W32 [33]	CVPR 2019	256×192	28.5	7.1	74.4	**90.5**	**81.9**	-	-	78.9
PRTR [19]	CVPR 2021	384×288	57.2	21.6	73.1	89.4	79.8	68.8	80.4	79.8
PRTR [19]	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

Tokenpose_B [23], 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 [33] 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 [12] and 3DPW [35] 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 [7, 16, 24], 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) [4] and the MPJPE after Procrustes alignment (PA-MPJPE) [4] are reported on Human3.6M. Besides these, to evaluate the reconstructed mesh, the Mean Per Vertex Error (MPVE) is reported on 3DPW. Following [16, 18, 30], Human3.6M, MPI-INF-3DHP, COCO, 3DPW are used for mixed training. Following previous work [16] [24] [25], 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 [25].

For 3DPW, FeatER improves the MPJPE from 89.7 [13] to 88.4, the PA-MPJPE from 55.8 [6] to 54.5, and MPVE from 107.1 [13] to 105.6 without using 3DPW training set. When using 3DPW training set during training, FeatER also shows superior performance compared to MeshGraphormer [25]. Moreover, FeatER reduces the memory and computational costs significantly (only 5% of Params and 16% of MACs compared with MeshGraphormer [25]). Thus, FeatER is a much more time and resource-efficient model for HPE and HMR tasks with exceptional performance.

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 shrinks 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

Table 2. 3D Pose and Mesh performance comparison with SOTA methods on Human3.6M and 3DPW datasets. The reported Params and MACs of FeatER are computed from the entire pipeline. † indicates video-based methods. The result of HybrIK* is with predicted camera parameters and ResNet34 is used as the backbone.

				Human3.6M		3DPW		
Model	Year	Params (M)	MACs (G)	MPJPE↓	PA-MPJPE↓	MPJPE↓	PA-MPJPE↓	MPVE↓
SPIN [16]	ICCV 2019	-	-	62.5	41.1	96.9	59.2	116.4
VIBE † [15]	CVPR 2020	-	-	65.6	41.4	82.9	51.9	99.1
I2LMeshNet [30]	ECCV 2020	140.5	36.6	55.7	41.1	93.2	57.7	-
TCMR † [6]	CVPR 2021	-	-	62.3	41.1	95.0	55.8	111.5
HybrIK* [18]	CVPR 2021	27.6	12.7	57.3	36.2	75.3	**45.2**	87.9
ProHMR [17]	ICCV 2021	-	-	-	41.2	-	59.8	-
PyMAF [44]	ICCV 2021	45.2	10.6	57.7	40.5	92.8	58.9	110.1
METRO [24]	CVPR 2021	229.2	56.6	54.0	36.7	77.1	47.9	88.2
MeshGraphormer [25]	ICCV 2021	226.5	56.6	51.2	34.5	74.7	45.6	87.7
DSR [9]	ICCV 2021	-	-	60.9	40.3	85.7	51.7	99.5
TCFormer [42]	CVPR 2022	-	-	62.9	42.8	80.6	49.3	-
FastMETRO [5]	ECCV 2022	48.5	15.8	53.9	37.3	77.9	48.3	90.6
FeatER		11.4	8.8	**49.9**	**32.8**	**73.4**	45.9	**86.9**

requires fewer 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 requiring 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 Supplementary B.

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 makes 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 Supplementary C.

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 [30] and PyMAF [44] 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 Supplementary F.

5. Conclusion

In this paper, we present FeatER, a novel feature map-based transformer architecture for HPE and HMR. FeatER

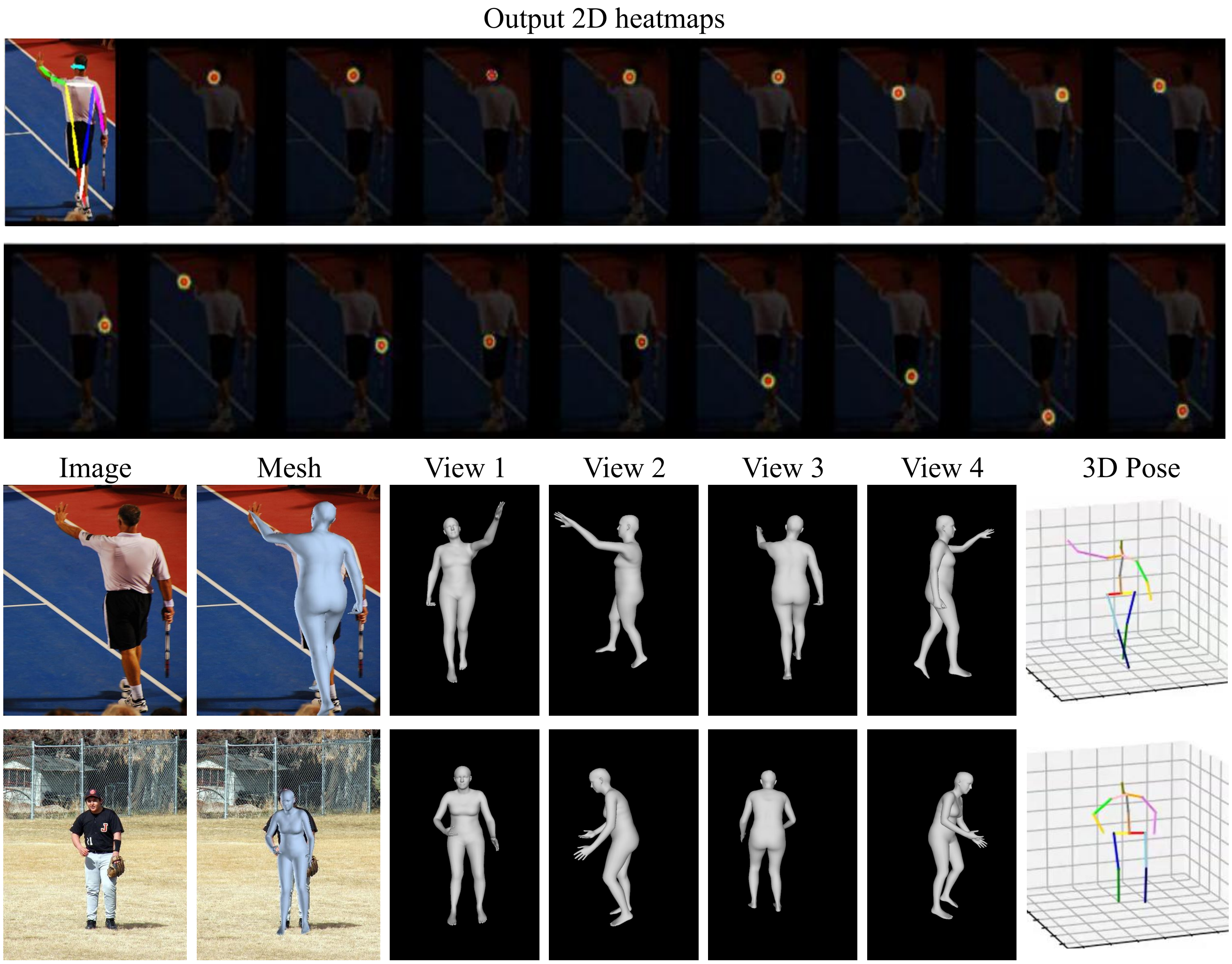

Figure 4. Qualitative results of the proposed FeatER. Images are taken from the in-the-wild COCO [26] dataset.

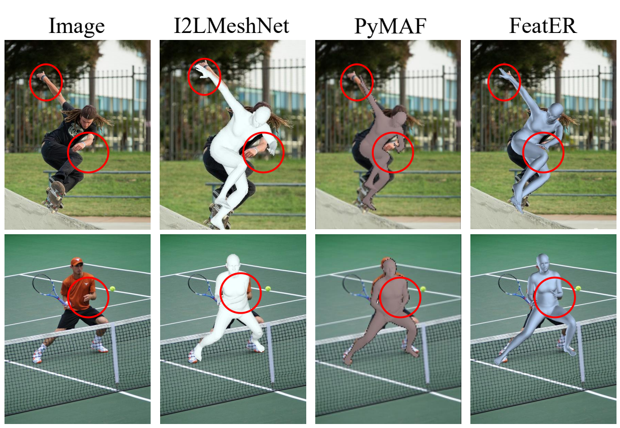

Figure 5. Qualitative comparison with other methods. Images are taken from the in-the-wild COCO [26] dataset. The red circles highlight locations where FeatER is more accurate than others. We follow previous work [24, 25, 30, 44] to visualize human mesh using the SMPL *gender neutral* model.

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 performance while significantly reducing the computational cost for HPE and HMR tasks.

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.

Acknowledgement: This work is supported by the National Science Foundation under Grant No. 1910844.

References

[1] 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*, pages 7291–7299, 2017. 1

[2] 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*, pages 213–229. Springer, 2020. 2

[3] 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*, pages 11030–11039, 2020. 6

[4] 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, 2020. 2, 5, 6

[5] 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. 7

[6] 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)*, pages 1964–1973, June 2021. 6, 7

[7] 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. 6

[8] 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. 2, 3

[9] 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*, pages 11250–11259, 2021. 3, 7

[10] 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

[11] 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)*, pages 15013–15022, October 2021. 2

[12] 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. 6

[13] 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. 1, 6

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[15] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020. 7

[16] 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. 6, 7

[17] 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*, pages 11605–11614, 2021. 7

[18] 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, 5, 6, 7

[19] 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*, pages 1944–1953, 2021. 2, 6

[20] 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*, 2021. 2

[21] 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. 2

[22] 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*, 2021. 2

[23] 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. 1, 2, 5, 6

[24] 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. 1, 2, 3, 6, 7, 8

[25] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021. 1, 2, 3, 6, 7, 8

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5, 8

[27] 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*, pages 2949–2958, 2021. 2

[28] 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)*, pages 116–131, 2018. 3

[29] 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*, pages 2906–2917, 2021. 2

[30] 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. 6, 7, 8

[31] 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. 5

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[33] 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, 3, 5, 6

[34] Yating Tian, Hongwen Zhang, Yebin Liu, and Limin Wang. Recovering 3d human mesh from monocular images: A survey. *arXiv preprint arXiv:2203.01923*, 2022. 2

[35] 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. 6

[36] 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)*, pages 466–481, 2018. 6

[37] 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*, pages 3601–3610, 2021. 2

[38] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11802–11812, 2021. 1, 2, 6

[39] 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. 3

[40] 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. 2, 6

[41] 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*, pages 12971–12980, 2021. 2

[42] 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*, pages 11101–11111, 2022. 7

[43] 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. 1

[44] 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. 7, 8

[45] 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*, pages 6848–6856, 2018. 3

[46] 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. 2, 3

[47] 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)*, pages 11656–11665, October 2021. 1, 2