Cross-Attention of Disentangled Modalities for 3D Human Mesh Recovery with Transformers

Junhyeong Cho[1] Kim Youwang[2] Tae-Hyun Oh[2,3,*]

[1]Department of CSE [2]Department of EE [3]Graduate School of AI
Pohang University of Science and Technology (POSTECH), Korea
{junhyeong99, youwang.kim, taehyun}@postech.ac.kr

https://github.com/postech-ami/FastMETRO

Abstract. Transformer encoder architectures have recently achieved state-of-the-art results on monocular 3D human mesh reconstruction, but they require a substantial number of parameters and expensive computations. Due to the large memory overhead and slow inference speed, it is difficult to deploy such models for practical use. In this paper, we propose a novel transformer encoder-decoder architecture for 3D human mesh reconstruction from a single image, called *FastMETRO*. We identify the performance bottleneck in the encoder-based transformers is caused by the token design which introduces high complexity interactions among input tokens. We disentangle the interactions via an encoder-decoder architecture, which allows our model to demand much fewer parameters and shorter inference time. In addition, we impose the prior knowledge of human body's morphological relationship via attention masking and mesh upsampling operations, which leads to faster convergence with higher accuracy. Our FastMETRO improves the Pareto-front of accuracy and efficiency, and clearly outperforms image-based methods on Human3.6M and 3DPW. Furthermore, we validate its generalizability on FreiHAND.

1 Introduction

3D human pose and shape estimation models aim to estimate 3D coordinates of human body joints and mesh vertices. These models can be deployed in a wide range of applications that require human behavior understanding, *e.g.*, human motion analysis and human-computer interaction. To utilize such models for practical use, monocular methods [2, 8, 15, 16, 20–22, 24, 25, 34, 36, 40, 44] estimate the 3D joints and vertices without using 3D scanners or stereo cameras. This task is essentially challenging due to complex human body articulation, and becomes more difficult by occlusions and depth ambiguity in monocular settings.

To deal with such challenges, state-of-the-art methods [24, 25] exploit non-local relations among human body joints and mesh vertices via transformer encoder architectures. This leads to impressive improvements in accuracy by consuming a substantial number of parameters and expensive computations as trade-offs; efficiency is less taken into account, although it is crucial in practice.

*Joint affiliated with Yonsei University, Korea.

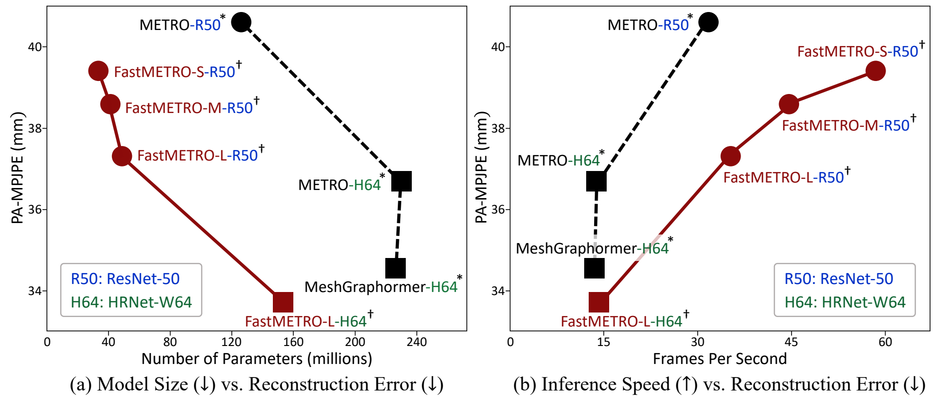


Fig. 1. Comparison with encoder-based transformers [24, 25] and our models on Human3.6M [14]. Our FastMETRO substantially improves the Pareto-front of accuracy and efficiency. † indicates training for 60 epochs, and $*$ denotes training for 200 epochs.

In this paper, we propose **FastMETRO** which employs a novel transformer encoder-decoder architecture for 3D human pose and shape estimation from an input image. Compared with the transformer encoders [24, 25], FastMETRO is more practical because it achieves competitive results with much fewer parameters and faster inference speed, as shown in Figure 1. Our architecture is motivated by the observation that the encoder-based methods overlook the importance of the token design which is a key-factor in accuracy and efficiency.

The encoder-based transformers [24, 25] share similar transformer encoder architectures. They take K joint and N vertex tokens as input for the estimation of 3D human body joints and mesh vertices, where K and N denote the number of joints and vertices in a 3D human mesh, respectively. Each token is constructed by the concatenation of a global image feature vector $\mathbf{x} \in \mathbb{R}^C$ and 3D coordinates of a joint or vertex in the human mesh. This results in the input tokens of dimension $\mathbb{R}^{(K+N)\times(C+3)}$ which are fed as input to the transformer encoders.[1] This token design introduces the same sources of the performance bottleneck: 1) spatial information is lost in the global image feature $\mathbf{x}$, and 2) the same image feature $\mathbf{x}$ is used in an overly-duplicated way. The former is caused by the average pooling operation to obtain the global image feature $\mathbf{x}$. The latter leads to considerable inefficiency, since expensive computations are required to process mostly duplicated information, where distinctively informative signals are only in 0.15% of the input tokens.[2] Furthermore, the computational complexity of each transformer layer is quadratic as $O(L^2C + LC^2)$, where $L \geq K + N$. Once either L or C is dominantly larger, it results in unfavorable efficiency. Both methods [24, 25] are such undesirable cases.

[1] For simplicity, we discuss the input tokens mainly based on METRO [24]. Mesh Graphormer [25] has subtle differences, but the essence of the bottleneck is shared.
[2] 3-dimensional coordinates out of $(C + 3)$-dimensional input tokens, where $C = 2048$.

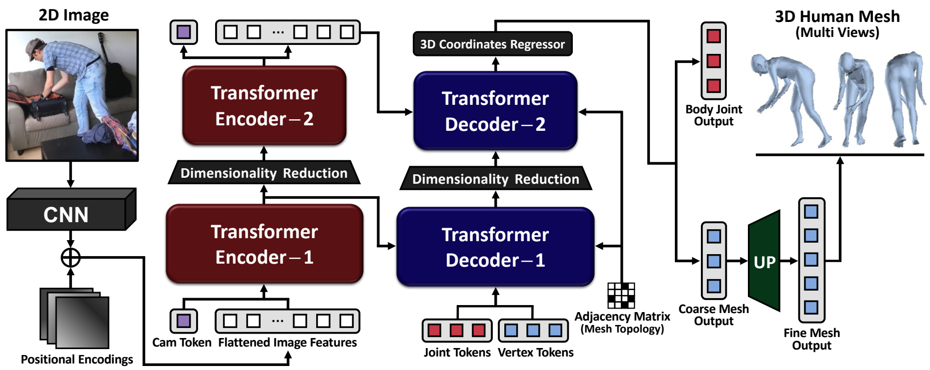


Fig. 2. Overall architecture of FastMETRO. Our model estimates 3D coordinates of human body joints and mesh vertices from a single image. We extract image features via a CNN backbone, which are fed as input to our transformer encoder. In addition to image features produced by the encoder, our transformer decoder takes learnable joint and vertex tokens as input. To effectively learn non-local joint-vertex relations and local vertex-vertex relations, we mask self-attentions of non-adjacent vertices according to the topology of human triangle mesh. Following [24, 25], we progressively reduce the hidden dimension sizes via linear projections in our transformer.

In contrast, our FastMETRO does not concatenate an image feature vector for the construction of input tokens. As illustrated in Figure 2, we disentangle the image encoding part and mesh estimation part via an encoder-decoder architecture. Our joint and vertex tokens focus on certain image regions through cross-attention modules in the transformer decoder. In this way, the proposed method efficiently estimates the 3D coordinates of human body joints and mesh vertices from a 2D image. To effectively capture non-local joint-vertex relations and local vertex-vertex relations, we mask self-attentions of non-adjacent vertices according to the topology of human triangle mesh. To avoid the redundancy caused by the spatial locality of human mesh vertices, we perform coarse-to-fine mesh upsampling as in [22, 24, 25]. By leveraging the prior knowledge of human body's morphological relationship, we substantially reduce optimization difficulty. This leads to faster convergence with higher accuracy.

We present the proposed method with model-size variants by changing the number of transformer layers: FastMETRO-S, FastMETRO-M, FastMETRO-L. Compared with the encoder-based transformers [24, 25], FastMETRO-S requires only about 9% of the parameters in the transformer architecture, but shows competitive results with much faster inference speed. In addition, the large variant (FastMETRO-L) achieves the state of the art on the Human3.6M [14] and 3DPW [32] datasets among image-based methods, which also demands fewer parameters and shorter inference time compared with the encoder-based methods. We demonstrate the effectiveness of the proposed method by conducting extensive experiments, and validate its generalizability by showing 3D hand mesh reconstruction results on the FreiHAND [47] dataset.

Our contributions are summarized as follows:

- We propose FastMETRO which employs a novel transformer encoder-decoder architecture for 3D human mesh recovery from a single image. Our method resolves the performance bottleneck in the encoder-based transformers, and improves the Pareto-front of accuracy and efficiency.
- The proposed model converges much faster by reducing optimization difficulty. Our FastMETRO leverages the prior knowledge of human body's morphological relationship, *e.g.*, masking attentions according to the human mesh topology.
- We present model-size variants of our FastMETRO. The small variant shows competitive results with much fewer parameters and faster inference speed. The large variant clearly outperforms existing image-based methods on the Human3.6M and 3DPW datasets, which is also more lightweight and faster.

2 Related Work

Our proposed method aims to estimate the 3D coordinates of human mesh vertices from an input image by leveraging the attention mechanism in the transformer architecture. We briefly review relevant methods in this section.

Human Mesh Reconstruction. The reconstruction methods belong to one of the two categories: parametric approach and non-parametric approach. The parametric approach learns to estimate the parameters of a human body model such as SMPL [29]. On the other hand, the non-parametric approach learns to directly regress the 3D coordinates of human mesh vertices. They obtain the 3D coordinates of human body joints via linear regression from the estimated mesh.

The reconstruction methods in the parametric approach [2,8,11,15,16,20,21,36,40,44] have shown stable performance in monocular 3D human mesh recovery. They have achieved the robustness to environment variations by exploiting the human body prior encoded in a human body model such as SMPL [29]. However, their regression targets are difficult for deep neural networks to learn; the pose space in the human body model is expressed by the 3D rotations of human body joints, where the regression of the 3D rotations is challenging [31].

Recent advances in deep neural networks have enabled the non-parametric approach with promising performance [6,22,24,25,34]. Kolotouros *et al.* [22] propose a graph convolutional neural network (GCNN) [18] to effectively learn local vertex-vertex relations, where the graph structure is based on the topology of SMPL human triangle mesh [29]. They extract a global image feature vector through a CNN backbone, then construct vertex embeddings by concatenating the image feature vector with the 3D coordinates of vertices in the human mesh. After iterative updates via graph convolutional layers, they estimate the 3D locations of human mesh vertices. To improve the robustness to partial occlusions, Lin *et al.* [24,25] propose transformer encoder architectures which effectively learn the non-local relations among human body joints and mesh vertices via the attention mechanism in the transformer. Their models, METRO [24] and Mesh Graphormer [25], follow the similar framework with the GCNN-based method [22]. They construct vertex tokens by attaching a global image feature vector to the 3D coordinates of vertices in the human mesh. After several updates via transformer encoder layers, they regress the 3D coordinates of human mesh vertices.

Among the reconstruction methods, METRO [24] and Mesh Graphormer [25] are the most relevant work to our FastMETRO. We found that the token design in those methods leads to a substantial number of unnecessary parameters and computations. In their architectures, transformer encoders take all the burdens to learn complex relations among mesh vertices, along with the highly non-linear mapping between 2D space and 3D space. To resolve this issue, we disentangle the image-encoding and mesh-estimation parts via an encoder-decoder architecture. This makes FastMETRO more lightweight and faster, and allows our model to learn the complex relations more effectively.

Transformers. Vaswani *et al.* [41] introduce a transformer architecture which effectively learns long-range relations through the attention mechanism in the transformer. This architecture has achieved impressive improvements in diverse computer vision tasks [3–5,7,12,17,24,25,27,28,30,38,43,45]. Dosovitskiy *et al.* [7] present a transformer encoder architecture, where a learnable token aggregates image features via self-attentions for image classification. Carion *et al.* [3] propose a transformer encoder-decoder architecture, where learnable tokens focus on certain image regions via cross-attentions for object detection. Those transformers have the most relevant architectures to our model.

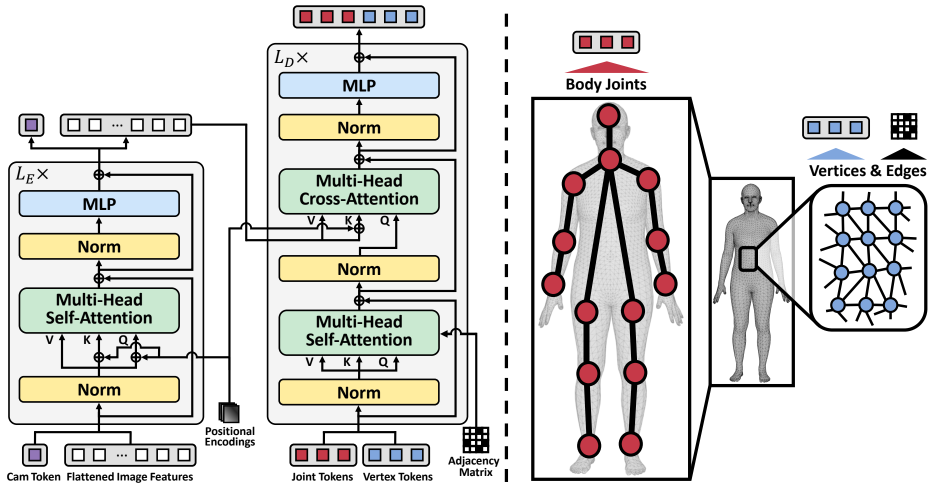
Our FastMETRO employs a transformer encoder-decoder architecture, whose decoupled structure is favorable to learn the complex relations between the heterogeneous modalities of 2D image and 3D mesh. Compared with the existing transformers [3–5, 7, 12, 17, 27, 30, 38, 43, 45], we progressively reduce hidden dimension sizes in the transformer architecture as in [24, 25]. Our separate decoder design enables FastMETRO to easily impose the human body prior by masking self-attentions of decoder input tokens, which leads to stable optimization and higher accuracy. This is novel in transformer architectures.

3 Method

We propose a novel method, called **Fast MEsh TRansfOrmer** (FastMETRO). FastMETRO has a transformer encoder-decoder architecture for 3D human mesh recovery from an input image. The overview of our method is shown in Figure 2. The details of our transformer encoder and decoder are illustrated in Figure 3.

3.1 Feature Extractor

Given a single RGB image, our model extracts image features $\mathbf{X}_I \in \mathbb{R}^{H \times W \times C}$ through a CNN backbone, where $H \times W$ denotes the spatial dimension size and C denotes the channel dimension size. A 1×1 convolution layer takes the image features $\mathbf{X}_I$ as input, and reduces the channel dimension size to D. Then, a flatten operation produces flattened image features $\mathbf{X}_F \in \mathbb{R}^{HW \times D}$. Note that we employ positional encodings for retaining spatial information in our transformer, as illustrated in Figure 3.



(a) Transformer Encoder-Decoder Architecture (b) Human Body Joints and Mesh Topology

Fig. 3. Details of our transformer architecture and 3D human body mesh. For simplicity, we illustrate the transformer without progressive dimensionality reduction. Note that the camera feature is not fed as input to the decoder. We mask attentions using the adjacency matrix obtained from the human triangle mesh of SMPL [29].

3.2 Transformer with Progressive Dimensionality Reduction

Following the encoder-based transformers [24, 25], FastMETRO progressively reduces the hidden dimension sizes in the transformer architecture via linear projections, as illustrated in Figure 2.

Transformer Encoder. Our transformer encoder (Figure 3a) takes a learnable camera token and the flattened image features $\mathbf{X}_F$ as input. The camera token captures essential features to predict weak-perspective camera parameters through the attention mechanism in the transformer; the camera parameters are used for fitting the 3D estimated human mesh to the 2D input image. Given the camera token and image features, the transformer encoder produces a camera feature and aggregated image features $\mathbf{X}_A \in \mathbb{R}^{HW \times D}$.

Transformer Decoder. In addition to the image features $\mathbf{X}_A$ obtained from the encoder, our transformer decoder (Figure 3a) takes the set of learnable joint tokens and the set of learnable vertex tokens as input. Each token in the set of joint tokens $\mathcal{T}_J = \{\mathbf{t}_1^J, \mathbf{t}_2^J, \ldots, \mathbf{t}_K^J\}$ is used to estimate 3D coordinates of a human body joint, where $\mathbf{t}_i^J \in \mathbb{R}^D$. The joint tokens correspond to the body joints in Figure 3b. Each token in the set of vertex tokens $\mathcal{T}_V = \{\mathbf{t}_1^V, \mathbf{t}_2^V, \ldots, \mathbf{t}_N^V\}$ is used to estimate 3D coordinates of a human mesh vertex, where $\mathbf{t}_j^V \in \mathbb{R}^D$. The vertex tokens correspond to the mesh vertices in Figure 3b. Given the image features and tokens, the transformer decoder produces joint features $\mathbf{X}_J \in \mathbb{R}^{K \times D}$ and vertex features $\mathbf{X}_V \in \mathbb{R}^{N \times D}$ through self-attention and cross-attention modules. Our transformer decoder effectively captures non-local relations among human body joints and mesh vertices via self-attentions, which improves the robustness to environment variations such as occlusions. Regarding the joint and vertex tokens, each focuses on its relevant image region via cross-attentions.

Attention Masking based on Human Mesh Topology. To effectively capture local vertex-vertex and non-local joint-vertex relations, we mask self-attentions of non-adjacent vertices according to the topology of human triangle mesh in Figure 3b. Although we mask the attentions of non-adjacent vertices, the coverage of each vertex token increases as it goes through decoder layers in the similar way with iterative graph convolutions. Note that GraphCMR [22] and Mesh Graphormer [25] perform graph convolutions based on the human mesh topology, which demands additional learnable parameters and computations.

3.3 Regressor and Mesh Upsampling

3D Coordinates Regressor. Our regressor takes the joint features $\mathbf{X}_J$ and vertex features $\mathbf{X}_V$ as input, and estimates the 3D coordinates of human body joints and mesh vertices. As a result, 3D joint coordinates $\hat{\mathbf{J}}_{\text{3D}} \in \mathbb{R}^{K\times 3}$ and 3D vertex coordinates $\hat{\mathbf{V}}_{\text{3D}} \in \mathbb{R}^{N\times 3}$ are predicted.

Coarse-to-Fine Mesh Upsampling. Following [22, 24, 25], our FastMETRO estimates a coarse mesh, then upsample the mesh. In this way, we avoid the redundancy caused by the spatial locality of human mesh vertices. As in [22], FastMETRO obtains the fine mesh output $\hat{\mathbf{V}}'_{\text{3D}} \in \mathbb{R}^{M\times 3}$ from the coarse mesh output $\hat{\mathbf{V}}_{\text{3D}}$ by performing matrix multiplication with the upsampling matrix $\mathbf{U} \in \mathbb{R}^{M\times N}$, *i.e.*, $\hat{\mathbf{V}}'_{\text{3D}} = \mathbf{U}\hat{\mathbf{V}}_{\text{3D}}$, where the upsampling matrix $\mathbf{U}$ is pre-computed by the sampling algorithm in [37].

3.4 Training FastMETRO

3D Vertex Regression Loss. To train our model for the regression of 3D mesh vertices, we use $L1$ loss function. This regression loss L^V_{3D} is computed by

$$L^V_{\text{3D}} = \frac{1}{M}\|\hat{\mathbf{V}}'_{\text{3D}} - \bar{\mathbf{V}}_{\text{3D}}\|_1, \tag{1}$$

where $\bar{\mathbf{V}}_{\text{3D}} \in \mathbb{R}^{M\times 3}$ denotes the ground-truth 3D vertex coordinates.

3D Joint Regression Loss. In addition to the estimated 3D joints $\hat{\mathbf{J}}_{\text{3D}}$, we also obtain 3D joints $\hat{\mathbf{J}}'_{\text{3D}} \in \mathbb{R}^{K\times 3}$ regressed from the fine mesh $\hat{\mathbf{V}}'_{\text{3D}}$, which is the common practice in the literature [6, 8, 16, 20–22, 24, 25, 34, 40, 44]. The regressed joints $\hat{\mathbf{J}}'_{\text{3D}}$ are computed by the matrix multiplication of the joint regression matrix $\mathbf{R} \in \mathbb{R}^{K\times M}$ and the fine mesh $\hat{\mathbf{V}}'_{\text{3D}}$, *i.e.*, $\hat{\mathbf{J}}'_{\text{3D}} = \mathbf{R}\hat{\mathbf{V}}'_{\text{3D}}$, where the regression matrix $\mathbf{R}$ is pre-defined in SMPL [29]. To train our model for the regression of 3D body joints, we use $L1$ loss function. This regression loss L^J_{3D} is computed by

$$L^J_{\text{3D}} = \frac{1}{K}(\|\hat{\mathbf{J}}_{\text{3D}} - \bar{\mathbf{J}}_{\text{3D}}\|_1 + \|\hat{\mathbf{J}}'_{\text{3D}} - \bar{\mathbf{J}}_{\text{3D}}\|_1), \tag{2}$$

where $\bar{\mathbf{J}}_{\text{3D}} \in \mathbb{R}^{K\times 3}$ denotes the ground-truth 3D joint coordinates.

Table 1. Configurations for the variants of FastMETRO. Each has the same transformer architecture with a different number of layers. Only transformer parts are described.

Model	#Params	Time (ms)	Enc–1 & Dec–1		Enc–2 & Dec–2	
			#Layers	Dimension	#Layers	Dimension
FastMETRO–S	9.2M	9.6	1	512	1	128
FastMETRO–M	17.1M	15.0	2	512	2	128
FastMETRO–L	24.9M	20.8	3	512	3	128

2D Joint Projection Loss. Following the literature [8, 16, 20–22, 24, 25, 40, 44], for the alignment between the 2D input image and the 3D reconstructed human mesh, we train our model to estimate weak-perspective camera parameters $\{s, \mathbf{t}\}$; a scaling factor $s \in \mathbb{R}$ and a 2D translation vector $\mathbf{t} \in \mathbb{R}^2$. The weak-perspective camera parameters are estimated from the camera feature obtained by the transformer encoder. Using the camera parameters, we get 2D body joints via an orthographic projection of the estimated 3D body joints. The projected 2D body joints are computed by

$$\hat{\mathbf{J}}_{\text{2D}} = s\Pi(\hat{\mathbf{J}}_{\text{3D}}) + \mathbf{t}, \tag{3}$$

$$\hat{\mathbf{J}}'_{\text{2D}} = s\Pi(\hat{\mathbf{J}}'_{\text{3D}}) + \mathbf{t}, \tag{4}$$

where $\Pi(\cdot)$ denotes the orthographic projection; $\left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix}\right]^\intercal \in \mathbb{R}^{3\times 2}$ is used for this projection in FastMETRO. To train our model with the projection of 3D body joints onto the 2D image, we use $L1$ loss function. This projection loss L^J_{2D} is computed by

$$L^J_{\text{2D}} = \frac{1}{K}(\|\hat{\mathbf{J}}_{\text{2D}} - \bar{\mathbf{J}}_{\text{2D}}\|_1 + \|\hat{\mathbf{J}}'_{\text{2D}} - \bar{\mathbf{J}}_{\text{2D}}\|_1), \tag{5}$$

where $\bar{\mathbf{J}}_{\text{2D}} \in \mathbb{R}^{K\times 2}$ denotes the ground-truth 2D joint coordinates.

Total Loss. Following the literature [6, 8, 16, 19–22, 24, 25, 34, 40, 44], we train our model with multiple 3D and 2D training datasets to improve its accuracy and robustness. This total loss L_{total} is computed by

$$L_{\text{total}} = \alpha(\lambda^V_{\text{3D}} L^V_{\text{3D}} + \lambda^J_{\text{3D}} L^J_{\text{3D}}) + \beta\lambda^J_{\text{2D}} L^J_{\text{2D}}, \tag{6}$$

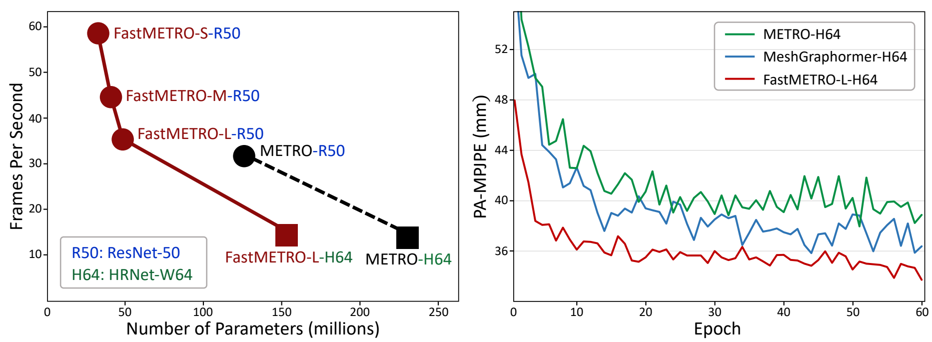
where $\lambda^V_{\text{3D}}, \lambda^J_{\text{3D}}, \lambda^J_{\text{2D}} > 0$ are loss coefficients and $\alpha, \beta \in \{0, 1\}$ are binary flags which denote the availability of ground-truth 3D and 2D coordinates.

4 Implementation Details

We implement our proposed method with three variants: FastMETRO-S, FastMETRO-M, FastMETRO-L. They have the same architecture with a different number of layers in the transformer encoder and decoder. Table 1 shows the configuration for each variant. Our transformer encoder and decoder are initialized with Xavier Initialization [9]. Please refer to the supplementary material for complete implementation details.

Table 2. Comparison with transformers for monocular 3D human mesh recovery on Human3.6M [14]. † and ∗ indicate training for 60 epochs and 200 epochs, respectively.

Model	CNN Backbone		Transformer		Overall		
	#Params	Time (ms)	#Params	Time (ms)	#Params	FPS	PA-MPJPE ↓
METRO–R50* [24]	23.5M	7.5	102.3M	24.2	125.8M	31.5	40.6
METRO–H64† [24]	128.1M	49.0	102.3M	24.2	230.4M	13.7	38.0
METRO–H64* [24]	128.1M	49.0	102.3M	24.2	230.4M	13.7	36.7
MeshGraphormer–H64† [25]	128.1M	49.0	98.4M	24.5	226.5M	13.6	35.8
MeshGraphormer–H64* [25]	128.1M	49.0	98.4M	24.5	226.5M	13.6	34.5
FastMETRO–S–R50†	23.5M	7.5	**9.2M**	**9.6**	**32.7M**	**58.5**	39.4
FastMETRO–M–R50†	23.5M	7.5	17.1M	15.0	40.6M	44.4	38.6
FastMETRO–L–R50†	23.5M	7.5	24.9M	20.8	48.4M	35.3	37.3
FastMETRO–L–H64†	128.1M	49.0	24.9M	20.8	153.0M	14.3	**33.7**



(a) Model Size (↓) vs. Inference Speed (↑) (b) Convergence Curve

Fig. 4. Comparison with encoder-based transformers [24, 25] and our proposed models on Human3.6M [14]. The small variant of our FastMETRO shows much faster inference speed, and its large variant converges faster than the transformer encoders.

5 Experiments

5.1 Datasets

Following the encoder-based transformers [24, 25], we train our FastMETRO with **Human3.6M** [14], **UP-3D** [23], **MuCo-3DHP** [33], **COCO** [26] and **MPII** [1] training datasets, and evaluate the model on P2 protocol in Human3.6M. Then, we fine-tune our model with **3DPW** [32] training dataset, and evaluate the model on its test dataset.

Following the common practice [6, 24, 25, 34], we employ the pseudo 3D human mesh obtained by SMPLify-X [35] to train our model with Human3.6M [14]; there is no available ground-truth 3D human mesh in the Human3.6M training dataset due to the license issue. For fair comparison, we employ the ground-truth 3D human body joints in Human3.6M during the evaluation of our model. Regarding the experiments on 3DPW [32], we use its training dataset for fine-tuning our model as in the encoder-based transformers [24, 25].

Table 3. Comparison with the state-of-the-art monocular 3D human pose and mesh recovery methods on 3DPW [32] and Human3.6M [14] among image-based methods.

Model	3DPW MPVPE ↓	3DPW MPJPE ↓	3DPW PA-MPJPE ↓	Human3.6M MPJPE ↓	Human3.6M PA-MPJPE ↓
HMR–R50 [16]	–	130.0	76.7	88.0	56.8
GraphCMR–R50 [22]	–	–	70.2	–	50.1
SPIN–R50 [21]	116.4	96.9	59.2	62.5	41.1
I2LMeshNet–R50 [34]	–	93.2	57.7	55.7	41.1
PyMAF–R50 [44]	110.1	92.8	58.9	57.7	40.5
ROMP–R50 [40]	105.6	89.3	53.5	–	–
ROMP–H32 [40]	103.1	85.5	53.3	–	–
PARE–R50 [20]	99.7	82.9	52.3	–	–
METRO–R50 [24]	–	–	–	56.5	40.6
DSR–R50 [8]	99.5	85.7	51.7	60.9	40.3
METRO–H64 [24]	88.2	77.1	47.9	54.0	36.7
PARE–H32 [20]	88.6	74.5	46.5	–	–
MeshGraphormer–H64 [25]	87.7	74.7	45.6	**51.2**	34.5
FastMETRO–S–R50	91.9	79.6	49.3	55.7	39.4
FastMETRO–M–R50	91.2	78.5	48.4	55.1	38.6
FastMETRO–L–R50	90.6	77.9	48.3	53.9	37.3
FastMETRO–L–H64	**84.1**	**73.5**	**44.6**	52.2	**33.7**

5.2 Evaluation Metrics

We evaluate our FastMETRO using three evaluation metrics: MPJPE [14], PA-MPJPE [46], MPVPE [36]. The unit of each metric is millimeter.
MPJPE. This metric denotes Mean-Per-Joint-Position-Error. It measures the Euclidean distances between the predicted and ground-truth joint coordinates.
PA-MPJPE. This metric is often called *Reconstruction Error*. It measures MPJPE after 3D alignment using Procrustes Analysis (PA) [10].
MPVPE. This metric denotes Mean-Per-Vertex-Position-Error. It measures the Euclidean distances between the predicted and ground-truth vertex coordinates.

5.3 Experimental Results

We evaluate the model-size variants of our FastMETRO on the 3DPW [32] and Human3.6M [14] datasets. In this paper, the inference time is measured using a single NVIDIA V100 GPU with a batch size of 1.
Comparison with Encoder-Based Transformers. In Table 2, we compare our models with METRO [24] and Mesh Graphormer [25] on the Human3.6M [14] dataset. Note that encoder-based transformers [24, 25] are implemented with ResNet-50 [13] (**R50**) or HRNet-W64 [42] (**H64**). FastMETRO-S outperforms METRO when both models employ the same CNN backbone (R50), although our model demands only 8.99% of the parameters in the transformer architecture. Regarding the overall inference speed, our model is 1.86× faster. It is worth noting that FastMETRO-L-R50 achieves similar results with METRO-H64, but our model is 2.58× faster. FastMETRO-L outperforms Mesh Graphormer when both models employ the same CNN backbone (H64), while our model demands only 25.30% of the parameters in the transformer architecture. Also, our model converges much faster than the encoder-based methods as shown in Figure 4.

Fig. 5. Qualitative results of our FastMETRO on Human3.6M [14] and 3DPW [32]. We visualize the 3D human mesh estimated by FastMETRO-L-H64. By leveraging the attention mechanism in the transformer, our model is robust to partial occlusions.

Comparison with Image-Based Methods. In Table 3, we compare our FastMETRO with the image-based methods for 3D human mesh reconstruction on 3DPW [32] and Human3.6M [14]. Note that existing methods are implemented with ResNet-50 [13] (**R50**) or HRNet-W32 [42] (**H32**) or HRNet-W64 [42] (**H64**). When all models employ R50 as their CNN backbones, FastMETRO-S achieves the best results without iterative fitting procedures or test-time optimizations. FastMETRO-L-H64 achieves the state of the art in every evaluation metric on the 3DPW dataset and PA-MPJPE metric on the Human3.6M dataset.

Visualization of Self-Attentions. In Figure 6, the first and second rows show the visualization of the attention scores in self-attentions between a specified body joint and mesh vertices. We obtain the scores by averaging attention scores from all attention heads of all multi-head self-attention modules in our transformer decoder. As shown in Figure 6, our FastMETRO effectively captures the non-local relations among joints and vertices via self-attentions in the transformer. This improves the robustness to environment variations such as occlusions.

Visualization of Cross-Attentions. In Figure 6, the third and fourth rows show the visualization of the attention scores in cross-attentions between a specified body joint and image regions. We obtain the scores by averaging attention scores from all attention heads of all multi-head cross-attention modules in our transformer decoder. As shown in Figure 6, the input tokens used in our

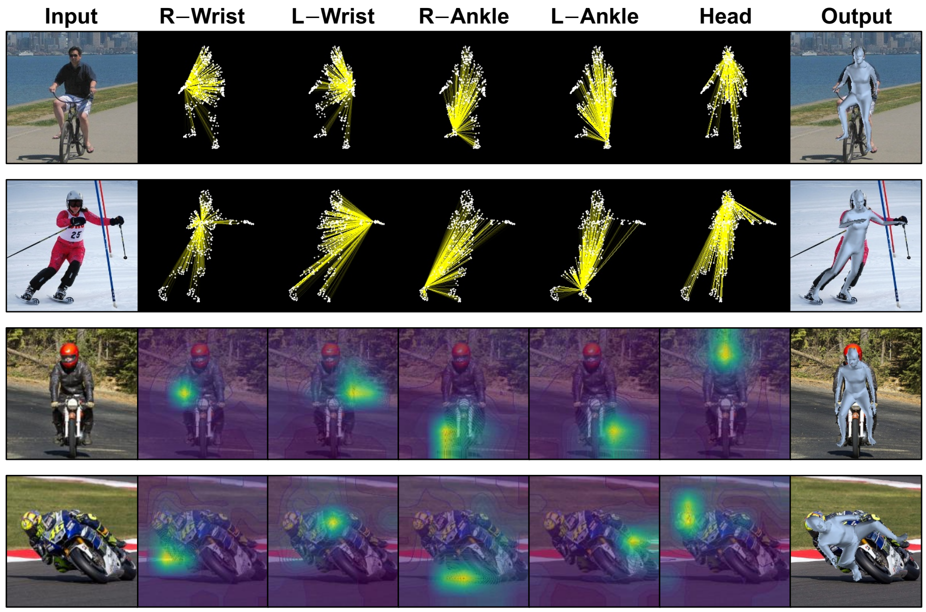


Fig. 6. Qualitative results of FastMETRO-L-H64 on COCO [26]. We visualize the attentions scores in self-attentions (top two rows) and cross-attentions (bottom two rows). The brighter lines or regions indicate higher attention scores.

transformer decoder focus on their relevant image regions. By leveraging the cross-attentions between disentangled modalities, our FastMETRO effectively learns to regress the 3D coordinates of joints and vertices from a 2D image.

5.4 Ablation Study

We analyze the effects of different components in our FastMETRO as shown in Table 4. Please refer to the supplementary material for more experiments.

Attention Masking. To effectively learn the local relations among mesh vertices, GraphCMR [22] and Mesh Graphormer [25] perform graph convolutions based on the topology of SMPL human triangle mesh [29]. For the same goal, we mask self-attentions of non-adjacent vertices according to the topology. When we evaluate our model without masking the attentions, the regression accuracy drops as shown in the first row of Table 4. This demonstrates that masking the attentions of non-adjacent vertices is effective. To compare the effects of attention masking with graph convolutions, we train our model using graph convolutions without masking the attentions. As shown in the second row of Table 4, we obtain similar results but this requires more parameters. We also evaluate our model when we mask the attentions in half attention heads, *i.e.*, there is no attention masking in other half attention heads. In this case, we get similar results using the same number of parameters as shown in the third row of Table 4.

Table 4. Ablation study of our FastMETRO on Human3.6M [14]. The effects of different components are evaluated. The default model is FastMETRO-S-R50.

Model	#Params	Human3.6M MPJPE ↓	Human3.6M PA-MPJPE ↓
w/o attention masking	**32.7M**	58.0	40.7
w/o attention masking + w/ graph convolutions	33.1M	56.6	**39.4**
w/ attention masking in half attention heads	**32.7M**	55.8	**39.4**
w/ learnable upsampling layers	45.4M	58.1	41.1
w/o progressive dimensionality reduction	39.5M	**55.5**	39.6
FastMETRO–S–R50	**32.7M**	55.7	**39.4**

Coarse-to-Fine Mesh Upsampling. The existing transformers [24,25] also first estimate a coarse mesh, then upsample the mesh to obtain a fine mesh. They employ two learnable linear layers for the upsampling. In our FastMETRO, we use the pre-computed upsampling matrix $\mathbf{U}$ to reduce optimization difficulty as in [22]; this upsampling matrix is a sparse matrix which has only about $25K$ non-zero elements. When we perform the mesh upsampling using learnable linear layers instead of the matrix $\mathbf{U}$, the regression accuracy drops as shown in the fourth row of Table 4, although it demands much more parameters.

Progressive Dimensionality Reduction. Following the existing transformer encoders [24,25], we also progressively reduce the hidden dimension sizes in our transformer via linear projections. To evaluate its effectiveness, we train our model using the same number of transformer layers but without progressive dimensionality reduction, *i.e.*, hidden dimension sizes in all transformer layers are the same. As shown in the fifth row of Table 4, we obtain similar results but this requires much more parameters. This demonstrates that the dimensionality reduction is helpful for our model to achieve decent results using fewer parameters.

Generalizability. Our model can reconstruct any arbitrary 3D objects by changing the number of input tokens used in the transformer decoder. Note that we can employ learnable layers for coarse-to-fine mesh upsampling without masking attentions. For 3D hand mesh reconstruction, there is a pre-computed upsampling matrix and a human hand model such as MANO [39]. Thus, we can leverage the matrix for mesh upsampling and mask self-attentions of non-adjacent vertices in the same way with 3D human mesh recovery. As illustrated in Figure 7, we can obtain an adjacency matrix and construct joint and vertex tokens from the human hand mesh topology. To validate the generalizability of our method, we train FastMETRO-L-H64 on the FreiHAND [47] training dataset and evaluate the model. As shown in Table 5, our proposed model achieves competitive results on FreiHAND.

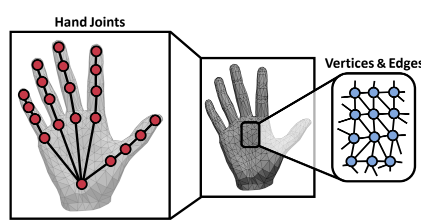


Fig. 7. Hand Joints and Mesh Topology.

Table 5. Comparison with transformers for monocular 3D hand mesh recovery on FreiHAND [14]. Test-time augmentation is not applied to these transformers.

Model	Transformer #Params	Overall #Params	FreiHAND PA-MPJPE ↓	FreiHAND F@15mm ↑
METRO–H64 [24]	102.3M	230.4M	6.8	0.981
FastMETRO–L–H64	**24.9M**	**153.0M**	**6.5**	**0.982**

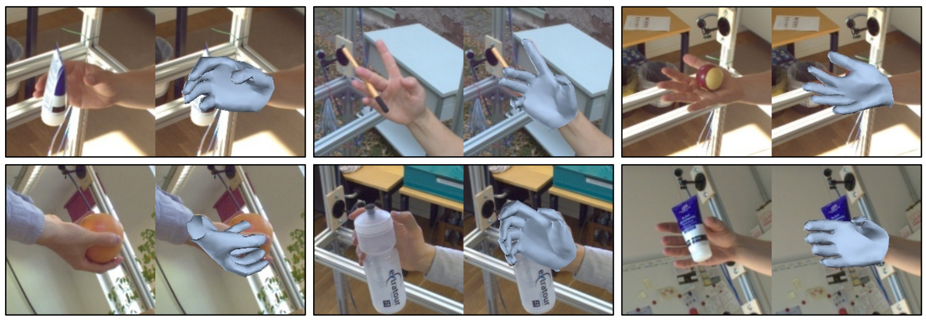


Fig. 8. Qualitative results of our FastMETRO on FreiHAND [47]. We visualize the 3D hand mesh estimated by FastMETRO-L-H64. By leveraging the attention mechanism in the transformer, our model is robust to partial occlusions.

6 Conclusion

We identify the performance bottleneck in the encoder-based transformers is due to the design of input tokens, and resolve this issue via an encoder-decoder architecture. This allows our model to demand much fewer parameters and shorter inference time, which is more appropriate for practical use. The proposed method leverages the human body prior encoded in SMPL human mesh, which reduces optimization difficulty and leads to faster convergence with higher accuracy. To be specific, we mask self-attentions of non-adjacent vertices and perform coarse-to-fine mesh upsampling. We demonstrate that our method improves the Pareto-front of accuracy and efficiency. Our FastMETRO achieves the robustness to occlusions by capturing non-local relations among body joints and mesh vertices, which outperforms image-based methods on the Human3.6M and 3DPW datasets. A limitation is that a substantial number of samples are required to train our model as in the encoder-based transformers.

Acknowledgments. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00290, Visual Intelligence for Space-Time Understanding and Generation based on Multi-layered Visual Common Sense; and No. 2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH)).

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014) 9
2. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: Keep it SMPL: Automatic Estimation of 3D Human Pose and Shape from a Single Image. In: Proceedings of the European Conference on Computer Vision (ECCV) (2016) 1, 4
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-End Object Detection with Transformers. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 213–229 (2020) 5
4. Cho, J., Yoon, Y., Kwak, S.: Collaborative Transformers for Grounded Situation Recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 5
5. Cho, J., Yoon, Y., Lee, H., Kwak, S.: Grounded Situation Recognition with Transformers. In: Proceedings of the British Machine Vision Conference (BMVC) (2021) 5
6. Choi, H., Moon, G., Lee, K.M.: Pose2Mesh: Graph Convolutional Network for 3D Human Pose and Mesh Recovery from a 2D Human Pose. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020) 4, 7, 8, 9
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations (ICLR) (2021) 5
8. Dwivedi, S.K., Athanasiou, N., Kocabas, M., Black, M.J.: Learning to Regress Bodies from Images using Differentiable Semantic Rendering. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 1, 4, 7, 8, 10
9. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 249–256 (2010) 8
10. Gower, J.C.: Generalized procrustes analysis. Psychometrika **40**, 33–51 (1975) 10
11. Guan, P., Weiss, A., Balan, A.O., Black, M.J.: Estimating Human Shape and Pose from a Single Image. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2009) 4
12. Guo, L., Liu, J., Zhu, X., Yao, P., Lu, S., Lu, H.: Normalized and Geometry-Aware Self-Attention Network for Image Captioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 5
13. He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016) 10, 11
14. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **36**(7), 1325–1339 (2014) 2, 3, 9, 10, 11, 13, 14
15. Jiang, W., Kolotouros, N., Pavlakos, G., Zhou, X., Daniilidis, K.: Coherent Reconstruction of Multiple Humans from a Single Image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 1, 4

16. Kanazawa, A., Black, M.J., Jacobs, D.W., Malik, J.: End-to-end Recovery of Human Shape and Pose. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 1, 4, 7, 8, 10
17. Kim, B., Lee, J., Kang, J., Kim, E.S., Kim, H.J.: HOTR: End-to-End Human-Object Interaction Detection with Transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 74–83 (2021) 5
18. Kipf, T.N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. In: International Conference on Learning Representations (ICLR) (2017) 4
19. Kocabas, M., Athanasiou, N., Black, M.J.: VIBE: Video Inference for Human Body Pose and Shape Estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 8
20. Kocabas, M., Huang, C.H.P., Hilliges, O., Black, M.J.: PARE: Part Attention Regressor for 3D Human Body Estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 1, 4, 7, 8, 10
21. Kolotouros, N., Pavlakos, G., Black, M.J., Daniilidis, K.: 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 (ICCV) (2019) 1, 4, 7, 8, 10
22. Kolotouros, N., Pavlakos, G., Daniilidis, K.: Convolutional Mesh Regression for Single-Image Human Shape Reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1, 3, 4, 7, 8, 10, 12, 13
23. Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M.J., Gehler, P.V.: Unite the People: Closing the Loop Between 3D and 2D Human Representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 9
24. Lin, K., Wang, L., Liu, Z.: End-to-End Human Pose and Mesh Reconstruction with Transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1954–1963 (2021) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14
25. Lin, K., Wang, L., Liu, Z.: Mesh Graphormer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 12939–12948 (2021) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 13
26. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft COCO: Common Objects in Context. In: Proceedings of the European Conference on Computer Vision (ECCV) (2014) 9, 12
27. Liu, S., Lin, T., He, D., Li, F., Deng, R., Li, X., Ding, E., Wang, H.: Paint Transformer: Feed Forward Neural Painting with Stroke Prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6598–6607 (2021) 5
28. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 5
29. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A Skinned Multi-Person Linear Model. ACM Transactions on Graphics (SIGGRAPH Asia) **34**(6), 248 (2015) 4, 6, 7, 12

30. Lu, Y., Rai, H., Chang, J., Knyazev, B., Yu, G., Shekhar, S., Taylor, G.W., Volkovs, M.: Context-Aware Scene Graph Generation with Seq2Seq Transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15931–15941 (2021) 5
31. Mahendran, S., Ali, H., Vidal, R.: A Mixed Classification-Regression Framework for 3D Pose Estimation from 2D Images. In: Proceedings of the British Machine Vision Conference (BMVC) (2018) 4
32. Marcard, T.v., Henschel, R., Black, M.J., Rosenhahn, B., Pons-Moll, G.: Recovering Accurate 3D Human Pose in The Wild Using IMUs and a Moving Camera. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018) 3, 9, 10, 11
33. Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Sridhar, S., Pons-Moll, G., Theobalt, C.: Single-Shot Multi-Person 3D Pose Estimation from Monocular RGB. In: International Conference on 3D Vision (3DV) (2018) 9
34. Moon, G., Lee, K.M.: I2L-MeshNet: Image-to-Lixel Prediction Network for Accurate 3D Human Pose and Mesh Estimation from a Single RGB Image. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020) 1, 4, 7, 8, 9, 10
35. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive Body Capture: 3D Hands, Face, and Body from a Single Image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 9
36. Pavlakos, G., Zhu, L., Zhou, X., Daniilidis, K.: Learning to Estimate 3D Human Pose and Shape from a Single Color Image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 1, 4, 10
37. Ranjan, A., Bolkart, T., Sanyal, S., Black, M.J.: Generating 3D Faces using Convolutional Mesh Autoencoders. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018) 7
38. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In: Advances in Neural Information Processing Systems (NIPS) (2021) 5
39. Romero, J., Tzionas, D., Black, M.J.: Embodied Hands: Modeling and Capturing Hands and Bodies Together. ACM Transactions on Graphics, (Proc. SIGGRAPH Asia) **36**(6) (2017) 13
40. Sun, Y., Bao, Q., Liu, W., Fu, Y., Black, M.J., Mei, T.: Monocular, One-Stage, Regression of Multiple 3D People. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 1, 4, 7, 8, 10
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is All you Need. In: Advances in Neural Information Processing Systems (NIPS) (2017) 5
42. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B.: Deep High-Resolution Representation Learning for Visual Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2019) 10, 11
43. Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., Xia, H.: End-to-End Video Instance Segmentation with Transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8741–8750 (2021) 5
44. Zhang, H., Tian, Y., Zhou, X., Ouyang, W., Liu, Y., Wang, L., Sun, Z.: PyMAF: 3D Human Pose and Shape Regression with Pyramidal Mesh Alignment Feedback Loop. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 1, 4, 7, 8, 10

45. Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., Ding, Z.: 3D Human Pose Estimation with Spatial and Temporal Transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 5
46. Zhou, X., Zhu, M., Pavlakos, G., Leonardos, S., Derpanis, K.G., Daniilidis, K.: MonoCap: Monocular Human Motion Capture using a CNN Coupled with a Geometric Prior. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) (2018) 10
47. Zimmermann, C., Ceylan, D., Yang, J., Russell, B., Argus, M., Brox, T.: FreiHAND: A Dataset for Markerless Capture of Hand Pose and Shape from Single RGB Images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2019) 3, 13, 14