This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

PostoMETRO: Pose Token Enhanced Mesh Transformer for Robust 3D Human Mesh Recovery

Wendi Yang[1,2*] **Zihang Jiang**[1,2*] **Shang Zhao**[1,2] **S. Kevin Zhou**[1,2,3,4†]
[1]School of Biomedical Engineering, Division of Life Sciences and Medicine,
University of Science and Technology of China (USTC), Hefei Anhui, 230026, China
[2]Center for Medical Imaging, Robotics, Analytic Computing & Learning (MIRACLE),
Suzhou Institute for Advance Research, USTC, Suzhou Jiangsu, 215123, China
[3]Key Laboratory of Precision and Intelligent Chemistry, USTC, Hefei Anhui, 230026, China
[4]Key Laboratory of Intelligent Information Processing of Chinese Academy of
Sciences (CAS), Institute of Computing Technology, CAS
yangwendi@mail.ustc.edu.cn {jzh0103, shangzhao}@ustc.edu.cn s.kevin.zhou@gmail.com

Abstract

*With the recent advancements in single-image-based 3D human pose and shape estimation (3DHPSE), there is a growing amount of works that can achieve good results on standard benchmarks but struggle to yield accurate human mesh in extreme scenarios like occlusion. Previous works propose to leverage 2D poses to help 3D HPSE model improve performance under occlusion, but usually rely on manual design to integrate 2D poses and only aim for specific kinds of occlusion. In this paper, we present PostoMETRO (**Pos**e **to**ken enhanced **ME**sh **TR**ansf**O**rmer), which integrates 2D pose prior knowledge as **tokens** into transformers to improve model's performance under occlusion. Using a VQ-VAE-based pose tokenizer, we efficiently represent 2D poses as tokens and feed them to transformers together with image tokens. Subsequently, these tokens are queried by vertex and joint tokens to decode 3D coordinates of mesh vertices and human joints. Our proposed 2D poses integration strategy is manual-design-free and suitable for various kinds of occlusion. Experiments on both standard and occlusion-specific benchmarks demonstrate the effectiveness of PostoMETRO. Code will be made available*[1].

1. Introduction

The 3D human pose and shape estimation (3DHPSE) task has been increasingly valued in the field of computer vision, due to the growing number of downstream tasks, such as virtual humans, VR, and AR. Among these, due to the inherent challenges of camera calibration in outdoor scenes, the 3DHPSE task under a monocular camera has received extensive research attention. With the help of human mesh parametric models like SMPL [32] and advanced deep learning architecture like CNN [13, 47] and Transformer [10], current works [6, 16, 22, 29, 51] can achieve good results on standard benchmarks, but tend to produce inaccurate human meshes in the in-the-wild occlusion scenarios, such as target human body is occluded by persons out-of-interest***(person-occlusion)*** or random object***(object-occlusion)***, limiting their further application.

The reason for the degeneration of current 3DHPSE models in the in-the-wild scenarios is the domain gap be-

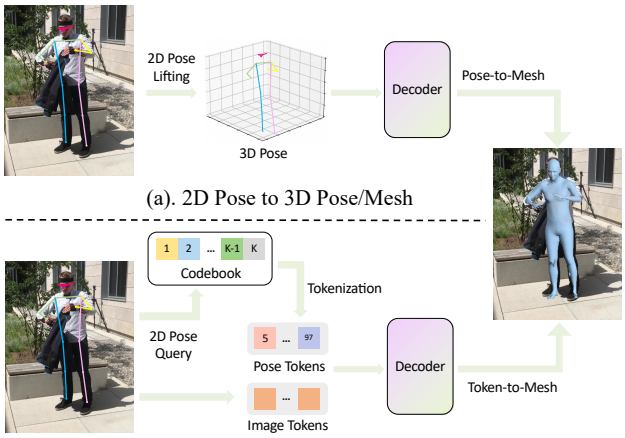


Figure 1. (a) Traditional approach of utilizing 2D pose information by converting it into 3D pose and subsequently generating a 3D mesh. (b) Transforming 2D pose into a sequence of tokens and integrating it with image tokens to generate a 3D mesh.

*These authors contributed equally to this work
†Corresponding author.
[1]https://github.com/PostoMETRO/PostoMETRO-Paper

tween training and testing. Due to its rich 3D annotations, the in-door datasets [14,35] takes up the majority of training datasets in many works [6, 16, 22, 29, 51]. However, the in-door images diverge from the testing-time in-the-wild images, since in-door images tend to be simpler and lack complex occlusion, leading to the training-testing difference. Therefore, the mainstream of solving occlusion problems in the 3DHPSE task is to mitigate such domain gap. Some work [21, 36, 50] introduce human pose prior like [38] or visibility hint to improve predictions. Another way [7, 8] is to use 2D poses from 2D pose estimators since it can offer human pose knowledge for 3DHPSE models, even under the in-the-wild scenes. [7] leverages 2D pose coordinates and converts the 3D HPSE task into 2 easier tasks(*i.e.*, 2D pose $\rightarrow$ 3D pose $\rightarrow$ 3D mesh, as shown in Fig. 1 (a)) and shows its effectiveness. Like [7], [8] uses 2D poses to calibrate CNN feature maps for better 3D HPSE results, showing the effectiveness of integrating 2D poses for solving person-occlusion problems.

Inspired by the idea of using 2D poses for occlusion, we present PostoMETRO, a novel paradigm to improve the performance of 3D HPSE model under occlusion scenarios, which is done by encoding 2D pose knowledge with images into transformer encoder-decoder in a cost-effective way. Different from [7,8] which need to construct 3D poses before obtaining human mesh, **we directly encode 2D human poses into tokens** by using a VQ-VAE, represented in Fig. 1 (b). By representing 2D poses in the form of tokens, PostoMETRO can directly use 2D pose knowledge in an end-to-end manner. Such integration makes PostoMETRO not limited to specific kind of occlusion, distinguishing it from previous works [8]. We then feed pose tokens and image tokens encoded by CNNs [13, 47] to **a transformer-based encoder**. To avoid the model being misled by inaccurate 2D poses, we incorporate the uncertainty from the 2D poses as part of pose tokens. To decode a human mesh from image and pose tokens, following [6, 29], we pre-define human mesh vertices and joints as tokens, **a transformer-based decoder** is used for interaction among these predefined tokens, image tokens, and pose tokens. A lightweight linear layer is used to transform joint and vertex features to 3D coordinates. The overall pipeline is in Fig. 2.

We benchmark our model on 3DPW [46], 3DPW-OCC [53], 3DPW-PC [43] and 3DOH [53] and get SOTA results, proving the proficiency of our model on various datasets and validating the effectiveness of our 2D pose integration strategy. Finally, we offer some ablation studies of our key design, showing the superiority of injecting 2D pose knowledge in the form of tokens and the effectiveness of using VQ-VAE to produce tokens.

In summary, our contributions are: i) We propose PostoMETRO, a novel framework to incorporate 2D pose into transformers for 3D HPSE task, which is done by transforming 2D pose as pose tokens. ii) Our method demonstrates competitive performance and occlusion-robustness on both standard benchmarks and occlusion-specific benchmarks. iii) Our approach can generalize to different occlusion scenarios including object-occlusion and person-occlusion without any dedicated design.

2. Related Work

2.1. 3D Human Pose and Shape Estimation

To estimate 3D human body, early works [3, 23, 39, 45] adopt optimization and use 2D keypoints and human silhouette to output parameters of parametric models [2, 32, 39, 48]. SMPLify [3] was the first method to fit the SMPL [32] model to the given 2D keypoints. These optimization-based methods are often slow and sensitive to initialization. Recently, with the advancement of neural networks, learning-based methods [6, 7, 16, 21, 22, 29, 51] are getting more and more popular. [16, 21, 22, 51] use image features to directly regress SMPL parameters. To facilitate training, extra supervisory signals are used, *e.g.*, 2D keypoints [16, 22] and human body segmentation map [21]. While parametric-model-based approaches have made remarkable progress, their performances are constrained by pre-defined body models, making it challenging to extend them to other applications. To tackle this, some works [6, 7, 29] proposed to use advanced neural network architecture [10, 19] for 3DHPSE task without relying on SMPL [32]. Choi *et al.* [7] use GCN [19] to predict 3D coordinates of mesh vertices, while Lin *et al.* [29] and Cho *et al.* [6] use vision transformer architecture [10] to predict 3D human mesh vertices.

2.2. Occlusion Handling

Solving the occlusion problem on single image can be very challenging. One straightforward way is performing data augmentation during training to synthesize occlusion [21, 26]. However, due to the texture difference between synthetic data and real occluders, such methods can not fully guarantee the performance of models in the real world. Zhang *et al.* [53] convert the occluded human shape estimation as a UV map inpainting problem, but it needs high-quality saliency maps and UV maps. Kocabas *et al.* [21] propose to infer unseen parts from visible parts, but it also relies on the model's capability to capture the high-quality visible parts. Similar to our work, Choi *et al.* [8] propose to leverage 2D human pose to mitigate the influence of occlusion. However, their approach only focuses on solving person-occlusion while our approach is suitable for object-occlusion and person-occlusion at the same time.

2.3. Structral Pose Representation

Using 2D pose for 3DHPSE tasks has been widely studied, but most works [15, 16, 22, 37] either just use 2D pose

for supervision or to estimate accurate SMPL parameters to help training. Choi *et al.* [8] adopt 2D pose to calibrate CNN and extract fine-grained features for improvements, but limited to person-occlusion scenarios. Lin *et al.* [28] use large-scale MoCap dataset [34] to generate 2D pose heatmaps and then pretrain transformers to recover human mesh by 2D pose token transformed from heatmaps. Their approach neglects 2D-to-3D ambiguity since it only uses 2D pose tokens as input, but 2D geometry hardly contains depth cues. Our approach can use image tokens to compensate for such ambiguity. Besides that, the scale of our pretraining datasets is more lightweight than that of MPT. Qiu *et al.* [40] combine token-level pose representation with image token to refine SMPL parameters. However, their approach aims for multi-person scenarios so that the pose token might contain pose information of multiple individuals, which brings complexities, while our approach does not have such problems since we focus on single-person scenarios. Closely related to our work, Geng *et al.* [11] propose to represent 2D poses using entries from VQ-VAE codebook. However, their work is proposed for 2D human pose regression. Constrained by the domain gap between 2D and 3D, their approach cannot be directly applied to 3DHPSE tasks.

3. Method

3.1. Preliminary: Pose Tokenizer

Following the progress in the field of image processing [41], we transform 2D pose into discrete token sequences by learning a "pose tokenizer". More specifically, a cropped human image $X \in \mathbb{R}^{C\times H\times W}$ is fed into the pose tokenizer, resulting in a sequence of human pose tokens $Z = [z_1, \ldots, z_N]^T \in \mathbb{R}^{N\times d_p}$ corresponding to the target human, where each token feature z_i is extracted from codebook $\mathbf{C} \in \mathbb{R}^{V\times d_p} = [c_1, ..., c_V]^T$.

We learn the pose tokenizer using a vector quantised-variational autoencoder (VQ-VAE) following [11]. To facilitate training, a two-stage training scheme is adopted. In stage one, an MLP-Mixer [44] based *pose encoder* is employed to compress 2D coordinates $P \in \mathbb{R}^{J\times 2}$ into a discrete token sequence $T = [t_1, \ldots, t_N] \in \mathbb{R}^{N\times d_p}$. A nearest-neighbor look-up is used to quantise each token t_i into token z_i in a codebook $\mathbf{C}$. Subsequently, an MLP-Mixer [44] based *decoder* is employed to decode the quantised token sequence Z back to 2D poses $\hat{P} \in \mathbb{R}^{J\times 2}$. Exponential moving average (EMA) is used to update the codebook due to the non-differentiable problem. In stage two, a *classifier* is trained to take a human image $X \in \mathbb{R}^{C\times H\times W}$ as input and output $\hat{L} \in \mathbb{R}^{N\times V}$ representing the likelihood of each token belonging to each entry z_i in the codebook $\mathbf{C}$. To explicitly model the uncertainty of 2D pose, we further extract the biggest logits from likelihood $\hat{L}$ as pose confidence $S_P \in \mathbb{R}^{N\times 1}$. Ground truth label $L \in \mathbb{R}^N$ can be obtained through the *pose encoder* trained in stage one and hence cross-entropy loss can be employed. Pose token sequence $\hat{T}_P \in \mathbb{R}^{N\times d_p}$ is computed by multiplication between $\hat{L}$ and $\mathbf{C}$. Pose tokenizer is trained on in-the-wild 2D pose datasets like [31]. Refer to [11] for more training and implementation details of the pose tokenizer.

We set the length of pose token sequence N to 34 and the number of entries in the codebook $\mathbf{C}$ to 2048. It is important to emphasize that the number of tokens in the sequence does not correspond to the number of human joints. In our work, we adopt the publicly available[2] pose tokenizer as described in [11]; note that *decoder* is not used since we focus on 3D human mesh recovery instead of 2D pose prediction.

3.2. Overall pipeline

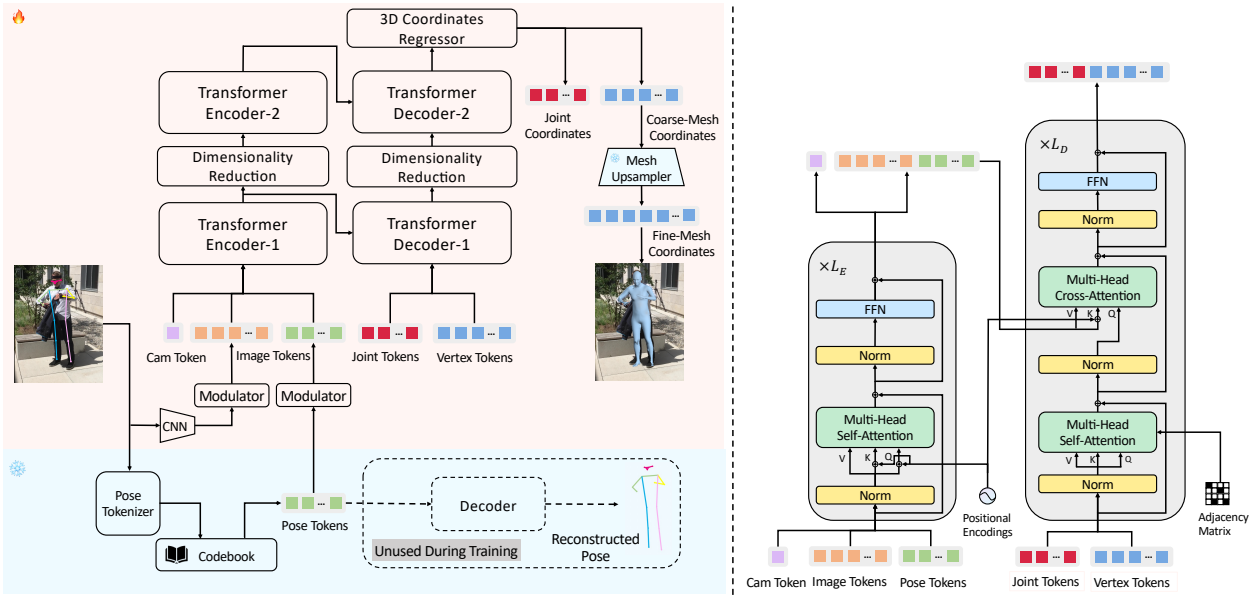
The overall pipeline of PostoMETRO is depicted in Fig. 2 (a). Our proposed method leverages two transformers to regress human mesh from single image. Transformer encoders are used to enable message passing between the camera token, image tokens, and pose tokens. The decoders extract useful information from the encoder, output joints features and vertex features, which will be decoded to human mesh. The encoder-decoder architecture is illustrated in Fig. 2 (b). An extra dimensionality reduction operation is conducted between transformers to reduce computations.

Feature Extraction Given a cropped human image $X \in \mathbb{R}^{C\times H\times W}$, a CNN backbone is employed to extract feature map $X_I \in \mathbb{R}^{d_i\times h\times w}$, note that d_i represents size of the channel dimension and $h \times w$ represents the spatial size. Typical values we use are $d_i = 2048$ and $h \times w = \frac{H}{32} \times \frac{W}{32}$. We concatenate pose token $\hat{T}_P$ and pose confidence S_P. Then, we utilize 4 MLP-Mixer-based [44] layers as the ***pose feature modulator*** to adjust the channel dimension size of their combination, resulting in a modulated sequence $T_P \in \mathbb{R}^{N\times d}$. Detailed architecture of our MLP-Mixer-based layer is provided in Supp. Mat.

Transformer Encoder First, a 1×1 convolution is used as the ***image feature modulator*** to convert the channel dimension of feature map X_I from d_i to d. We flatten and transpose the reduced feature map and obtain image token sequence $T_I \in \mathbb{R}^{hw\times d}$. Transformer encoder of PostoMETRO takes image tokens along with learnable camera token $T_C \in \mathbb{R}^{1\times d}$ and modulated pose tokens $T_P \in \mathbb{R}^{N\times d}$ as input, depicted in Fig. 2 (b). After that, transformer encoder outputs camera feature and image-pose feature $T_{IP} \in \mathbb{R}^{(hw+N)\times d}$. The obtained image-pose feature will be used as key and value in transformer decoder, while the camera feature will be solely decoded by a linear layer (not shown in Fig. 2), resulting in a scaling factor $s \in \mathbb{R}$ and a translation vector $t \in \mathbb{R}^2$, which can be used for loss computation.

Transformer Decoder Transformer decoder of PostoMETRO takes both joint tokens $\mathrm{T}_J \in \mathbb{R}^{K\times d}$ and vertex

[2] https://github.com/Gengzigang/PCT



(a) Overall Pipeline of PostoMETRO

(b) Transformer Encoder-Decoder Architecture

Figure 2. **Overall pipeline of PostoMETRO (a) and architecture of transformer encoder-decoder (b).** Before training transformers, we learn a "pose tokenizer" in VQ-VAE fashion, where a human image will be tokenized to discrete pose tokens, which can be reconstructed to a 2D pose by a decoder, according to the learned codebook. During training transformers, we freeze pose tokenizer and mesh upsampler and train transformers by concatenating pose token to camera token and image tokens as input for transformer encoders. For transformer decoders, learnable joint tokens and vertex tokens are fed as input. Following [6, 18], dimensionality reduction and adjacency matrix are used for effectiveness. Note that pose confidence for 2D pose is omitted for simplicity.

tokens $\mathrm{T}_V \in \mathbb{R}^{Q\times d}$ as input. Following [6, 18], we apply an attention mask derived from the adjacency matrix of human mesh vertices in the self-attention module. The output joint feature and vertex feature are then projected into 3D coordinates by a linear 3D coordinates regressor. As shown in Fig. 2 (a), a pre-defined mesh upsampler [42] $U \in \mathbb{R}^{M\times Q}$ is further applied to upsample the coarse mesh vertices $V_c \in \mathbb{R}^{Q\times 3}$ to fine mesh vertices $V_f \in \mathbf{R}^{M\times 3}$, *i.e.*, $V_f = UV_c$. We set $K = 14$, $Q = 431$ and $M = 6890$.

3.3. Loss Design

We train PostoMETRO with 3D vertex loss L_v, 3D joints loss L_{3D}, and 2D joints loss L_{2D}.

For 3D vertex loss L_v, L_1 loss is adopted and it could be represented as follows:

$$L_v = \frac{1}{M}\sum_{i=1}^{M}\|\hat{V}^i - V^i\|_1, \tag{1}$$

where $\hat{V}^i$ denotes the predicted ith vertex, and V^i represents the corresponding ground truth. M denotes the total number of human mesh vertices.

As for 3D joints loss L_{3D}, it is worth noting that in our experiments, predicted 3D coordinates of human joints come from two different sources: $\bar{J}_{3D}$ which is obtained by decoding the joint token using linear layer and $\hat{J}_{3D}$ which is obtained by applying pre-computed joint regression matrix M_J on human mesh $\hat{V}$, *i.e.*, $\hat{J}_{3D} = M_J\hat{V}$. This loss can be represented as follows:

$$L_{3D} = \frac{1}{K}\sum_{i=1}^{K}(\|\hat{J}^i_{3D} - J^i_{3D}\|_1 + \|\bar{J}^i_{3D} - J^i_{3D}\|_1), \tag{2}$$

where J_{3D} denotes the ground truth 3D coordinates of human joints. K denotes the total number of human joints.

Following previous works [16, 22], we use 2D joints loss L_{2D} to supervise model. We apply orthogonal projection operation $\Pi(\cdot)$ on 3D joints $\bar{J}_{3D}$, $\hat{J}_{3D}$ to obtain 2D joints $\bar{J}_{2D}$, $\hat{J}_{2D}$ coordinates by using camera parameters $\{s, t\}$ decoded from camera token as follows:

$$\hat{J}_{2D} = s\Pi(\hat{J}_{3D}) + t, \quad \bar{J}_{2D} = s\Pi(\bar{J}_{3D}) + t; \tag{3}$$

hence the 2D loss can be represented as follows:

$$L_{2D} = \frac{1}{K}\sum_{i=1}^{K}(\|\hat{J}^i_{2D} - J^i_{2D}\|_1 + \|\bar{J}^i_{2D} - J^i_{2D}\|_1). \tag{4}$$

To summarize, our overall loss can be represented as :

$$L_{total} = \alpha_1 L_v + \alpha_2 L_{3D} + \alpha_3 L_{2D}, \tag{5}$$

where α_1, α_2, α_3 are all weight terms, which are empirically set to 1000, 1000 and 100, respectively.

4. Experiments

4.1. Experimental Setup

Datasets. Following prior works [5, 6, 29], we adopt Human3.6M [14], UP-3D [23], MuCo-3DHP [35] as

3D datasets and pseudo-ground-truth SMPL annotated COCO [31] and MPII [1] as 2D datasets for training our model. For fair comparison, when testing on 3DPW-OCC [53], 3DPW-PC [43] and 3DPW-ALL [46], we do not fine-tune our model on any split of 3DPW [46] dataset. When testing on 3DOH [53], we fine-tune our model on 3DOH training set following [21, 24]. When testing on 3DPW-TEST [46] split, we fine-tune our model with 3DPW-TRAIN [46] split like [6, 28–30]. For each input image, the region of target human is cropped using annotated bounding boxes and resized to 256×256. We only apply color jittering, random rotation and scaling to images as data augmentation.

Implementation Details. All the experiments are implemented with PyTorch. We train our model on NVIDIA RTX3090 GPUs for 30 epochs. The total batch size is 128 when using ResNet-50 and 96 when using HRNet-W48. AdamW optimizer is adopted with its momentum factors set to 0.9 and 0.999. The learning rate is set to 2×10^{-5} during fine-tuning on the 3DPW-TRAIN split, and is set to 1×10^{-4} by default otherwise. Note that in all tables, if not otherwise stated, the results of our methods are obtained after fine-tuning on the training set.

Evaluation Metrics. We adopt three commonly used metrics: mean per vertex position error (**MPVPE**), mean per joint position error (**MPJPE**) and Procrustes-aligned mean per joint position error (**PA-MPJPE**). **MPVPE** measures the alignment between predicted human mesh vertices and its ground truth, while **MPJPE** and **PA-MPJPE** measure the accuracy of the recovered 3D human poses.

4.2. Main Results

Quantitative Results. We test our model on different datasets, *i.e.*, 3DPW-TEST, 3DPW-OCC [53], 3DOH [53] and 3DPW-PC [43] to validate its effectiveness. For better understanding, **we classify methods into three types based on 2D pose information exploitation.** First, image-based, which only explicitly uses image information. Second, 2D Pose-based, which only explicitly uses 2D pose information. Lastly, Mixed-based, which explicitly uses both.

We report the performance of PostoMETRO on 3DPW-TEST in Tab. 1. We fine-tune our model on 3DPW-TRAIN following prior works [6, 18, 29, 30]. As can be seen, our model with an HRNet-W48 backbone achieves SOTA performance on 3DPW-TEST, with 76.8mm MPVPE, 67.7mm MPJPE and 39.8mm PA-MPJPE. Despite that our method is indeed inferior to MPT [28] and VirtualMarker [33] on MPJPE, it beats all baselines significantly on MPVPE and PA-MPJPE, showing the capability of PostoMETRO on the in-the-wild dataset. It is worth noting that MPT benefits from a much larger pre-training dataset, which could explain the performance gap. For more details on dataset scale comparison, please see Supp. Mat. It is also worth noting that even when replacing our HRNet-W48 with a ResNet-50, compared to other methods using ResNet, our model still yields superior results.

To stay fair, we would like to point out some key distinctions between our baseline model, called FastMETRO, and our variant without pose tokens. Although both models share the same architecture, there's a notable difference in their results. The reasons are mainly threefold: 1. The resolution of images increases from 224×224 to 256×256. 2. The datasets we used are refined with more pseudo-ground-truth SMPL annotations. 3. We use backbones pretrained on COCO [31] rather than ImageNet [9]. For more details, please refer to Supp. Mat. We highlight that despite the extra improvements brought by the above, the benefits of pose tokens are still remarkable. When using ResNet-50, by adding pose tokens, PostoMETRO outperforms our w/o pose tokens variant by 5.5%, 5.5%, and 8.1% on MPVPE, MPJPE and PA-MPJPE, respectively. Even when comparing with a strong baseline, *i.e.*, HRNet-W48, the improvement still remains non-trival, particularly with a 4.1% performance gain on PA-MPJPE.

We also conduct experiments on occlusion datasets. On 3DPW-OCC, PostoMETRO with HRNet-W48 backbone outperforms other SOTA methods, including those targeted for occlusion scenarios, such as PARE [21] and NIKI [24]. On 3DOH, a similar superiority can also be observed. These results on different datasets show the robustness of our model under the challenge of object occlusion. On 3DPW-PC, PostoMETRO still shows competitive results. Despite that our method is inferior to 3DCrowdNet [8] on MPJPE, it still outperforms all methods on MPVPE and PA-MPJPE. Note that 3DCrowdNet also underperforms PostoMETRO on object-occlusion datasets in Tab. 2. Despite that we do not manually design any components for different occlusion(*i.e.*, object-occlusion or person-occlusion), our method can handle them equally well, distinguishing it from previous works [8]. **We also provide results of training and inference speed of our methods along with comparisons with baselines in Supp. Mat.**

Qualitative Results. We test FastMETRO [6] and our method on 3DPW [46] and OCHuman [52]. As in Fig. 3, we observe that PostoMETRO captures information about occluded body parts, such as limbs, as highlighted in the red rectangles. This shows the benefit of 2D pose tokens for enhanced 3D mesh reconstruction.

4.3. Ablation Studies

Effect of Different Tokens. We train our model with different types of tokens, *i.e.*, image tokens and pose tokens, and test each variant on 3DPW-OCC, 3DPW-PC, 3DPW-ALL and 3DPW-TEST. We offer statics of GPU memory usage and FPS of each variant in Tab. 4. Qualitative results in Tab. 4 also show that our proposed pose token

Methods	Paradigm	Backbone	3DPW-TEST		
			MPVPE(↓)	MPJPE(↓)	PA-MPJPE(↓)
HMR† [16]	Image-based	ResNet-50	-	130.0	81.3
SPIN† [22]	Image-based	ResNet-50	116.4	96.9	59.2
VIBE [20]	Image-based	ResNet-50	99.1	82.9	51.9
3DCrowdNet† [8]	Mixed-based	ResNet-50	98.3	81.7	51.5
ROMP [43]	Image-based	ResNet-50	94.7	79.7	49.7
HybrIK [25]	Image-based	ResNet-34	86.5	74.1	45.0
VisDB [50]	Image-based	ResNet-50	83.5	72.1	44.1
INT-2 [49]	Image-based	ResNet-50	87.9	75.6	42.0
Ours(w/o pose tokens)	Mixed-based	ResNet-50	82.5	72.4	44.4
Ours	Mixed-based	ResNet-50	**78.0**(↓5.5%)	**68.4**(↓5.5%)	**40.8**(↓8.1%)
Pose2Mesh [7]	2D Pose-based	HRNet-W48	105.3	89.5	56.3
PARE [21]	Image-based	HRNet-W32	88.6	74.5	46.5
METRO [29]	Image-based	HRNet-W64	88.2	77.1	47.9
MeshGraphormer [30]	Image-based	HRNet-W64	87.7	74.7	45.6
FastMETRO [6]	Image-based	HRNet-W64	84.1	72.5	44.6
PSVT [40]	Image-based	HRNet-W32	84.0	73.1	43.5
CLIFF [26]	Image-based	HRNet-W48	81.2	69.0	43.0
MPT [28]	2D Pose-based	HigherHRNet	79.4	**65.9**	42.8
VirtualMarker [33]	Image-based	HRNet-W48	77.9	67.5	41.3
NIKI [24]	Image-based	HRNet-W48	81.6	71.3	40.6
SMPLer-X-L32† [4]	Image-based	Vision Transformer	-	75.2	50.5
HMR2.0a† [12]	Image-based	Vision Transformer	-	70.0	44.5
OSX [27]	Image-based	Vision Transformer	-	74.7	45.1
Ours(w/o pose tokens)	Mixed-based	HRNet-W48	78.6	68.9	41.5
Ours	Mixed-based	HRNet-W48	**76.8**(↓2.3%)	67.7(↓1.7%)	**39.8**(↓4.1%)

Table 1. Performance on 3DPW [46]. Backbones are denoted for fair comparison. Results in bold indicate the best performance. † denotes the performance ***without*** fine-tuning on the 3DPW train-split. Compared to ours, MPT [28], SMPLer-X-L32 [4] and HMR2.0a [12] use additional datasets for training. Text in blue indicates the performance improvement compared with w/o pose tokens baseline.

Methods	Paradigm	3DPW-OCC			3DOH	
		MPVPE(↓)	MPJPE(↓)	PA-MPJPE(↓)	MPJPE(↓)	PA-MPJPE(↓)
Zhang *et al.* [53]	Image-based	-	-	72.2	-	58.5
HMR-EFT [15]	Image-based	111.3	94.4	60.9	75.2	53.1
PARE [21]	Image-based	107.9	90.5	56.6	63.3	44.3
3DCrowdNet [8]	Mixed-based	102.0	84.1	57.8	-	-
VisDB [50]	Image-based	110.5	87.3	56.0	60.9	42.7
NIKI [24]	Image-based	107.6	85.5	53.5	38.8	28.7
Ours (ResNet-50)*	Mixed-based	93.5	84.0	50.0	39.2	28.7
Ours (HRNet-W48)*	Mixed-based	**90.1**	**79.7**	**49.0**	**35.4**	**26.8**

Table 2. Performance on 3DPW-OCC [53] and 3DOH [53]. Note that all methods are tested ***without*** any fine-tuning on 3DPW dataset. Results in bold indicate the best performance. * means extra datasets used compared to PARE [21]. Note that we fine-tune our model on 3DOH **only** when testing on 3DOH.

enhanced model outperforms baselines by a large margin on each split. On the 3DPW-OCC dataset, using only image or pose tokens yields inferior results. However, combining both as input significantly improves performance. On the 3DPW-PC dataset, using only pose tokens outperforms models with only image tokens. Like on the 3DPW-OCC dataset, combining pose and image tokens further improves results. Such consistent benefits across various occlusion scenarios highlights the general applicability of our method, distinguishing it from approaches [8] aiming for person-occlusion. Such improvement can also be seen on the 3DPW-ALL and 3DPW-TEST splits, indicating that integrating 2D poses with images is useful for general 3D HPSE tasks.

Figure 3. **Qualitative results on 3DPW (rows 1-3) and OCHuman (rows 4-5) datasets.** From left to right: (a) Input image, (b) 2D Pose decoded from pose tokens, (c) FastMETRO [6] results, (d) Our results.

Methods	Paradigm	3DPW-PC		
		MPVPE(↓)	MPJPE(↓)	PA-MPJPE(↓)
VIBE	Image-based	-	-	103.5
SPIN [22]	Image-based	157.6	129.6	82.6
PyMaf [51]	Image-based	154.3	126.7	81.3
ROMP [43]	Image-based	147.5	115.6	75.8
OCHMR [17]	Image-based	145.9	112.2	75.2
3DCrowdNet [8]	Mixed-based	115.4	**91.3**	65.2
Ours (ResNet-50)	Mixed-based	113.4	97.8	62.4
Ours (HRNet-W48)	Mixed-based	**110.9**	95.3	**61.0**

Table 3. Performance on 3DPW-PC [43]. Note that all methods are tested ***without*** any fine-tuning on 3DPW dataset. Results in bold indicate the best performance.

Generalizability of Pose Tokens. We also show the generalizability of 2D pose tokens in 3DHPSE tasks. Representing 2D poses as tokens enables us to inject 2D pose knowledge into other architecture(e.g. METRO [29]) on-the-fly. Here we add 2D pose tokens to METRO by simply concatenating them to the original token sequence of METRO, training both it and the original METRO for 50 epochs. The results in Tab. 5c show the generalizability of 2D pose tokens, validating their ability to plug and play.

Effect of Tokenization. Unlike [7, 8] which use 2D poses as *coordinates*, we use 2D poses as *tokens*. Instead of using 2D poses as sets of independent joints, tokenizing 2D poses into tokens can not only store joint locations but also model high-level information like dependencies among joints. Here we follow [8] and sample image features from 2D joint locations. We replace pose tokens with these features for training, yet get inferior results, as shown in Tab. 5a, which can be attributed to the high-dimensional property of tokens, showing the effect of tokenization.

Benefits of VQ-VAE. The codebook of VQ-VAE explicitly stores 2D pose prior knowledge and benefit HMR model training **from start to end**. We also try other pose tokens,*e.g.*, embedding 2D pose coordinates from PCT [11] into tokens. We train this 2D-coords-variant using the same datasets for the same training time(including pretraining). The results in Tab. 5b show that such training might not guarantee effective encoding of 2D pose knowledge, validating the importance of VQ-VAE and its pretraining.

Design of Modulator. We ablate the choice of modulator in Tab. 5d. The best results are obtained when MLP-Mixer(*mixer*) is used for pose tokens and a 1×1 convolution layer(*linear*) is used for image tokens. Besides multipli-

Image Tokens	Pose Tokens	GPU memory	FPS	3DPW-OCC		3DPW-PC		3DPW-ALL		3DPW-TEST	
				MPJPE(↓)	PA-MPJPE(↓)	MPJPE(↓)	PA-MPJPE(↓)	MPJPE(↓)	PA-MPJPE(↓)	MPJPE(↓)	PA-MPJPE(↓)
	✓	∼2770MB	22.9FPS	90.2	56.4	104.1	65.7	93.3	54.5	96.8	55.6
✓		∼1920MB	34.9FPS	92.1	56.3	111.0	71.0	91.9	53.2	93.4	53.2
✓	✓	∼2890MB	18.7FPS	**84.0**	**50.0**	**97.8**	**62.4**	**83.1**	**48.2**	**84.9**	**48.9**

Table 4. GPU memory usage and FPS of different token-based-variant. Quantitative comparisons between these variants are also done on 3DPW-OCC [53], 3DPW-PC [43] and 3DPW-ALL [46]. Note that we use ResNet-50 as our backbone. Results are obtained ***without*** fine-tuning on 3DPW-TRAIN split and those in bold indicate the best performance.

Strategy	3DPW-TEST		
	MPVPE (↓)	MPJPE (↓)	PA-MPJPE (↓)
Feat sampled [8](w. R50)	100.7	91.9	51.9
Ours (w. R50)	**91.6**	**84.9**	**48.9**

(a) 2D Pose Knowledge Integration Strategy

Pose Tokens	3DPW-TEST		
	MPVPE (↓)	MPJPE (↓)	PA-MPJPE (↓)
2D coords (w. R50)	97.6	87.6	50.0
Ours (w. R50)	**91.6**	**84.9**	**48.9**

(b) Performance of Different Pose Tokens

Methods	3DPW-TEST	
	MPJPE (↓)	PA-MPJPE (↓)
METRO(w. R50)	102.1	60.0
METRO+pose(w. R50)	**96.8**	**58.3**

(c) Generalizability of 2D Pose Tokens

Image tokens	Pose tokens	3DPW-TEST	
		MPJPE(↓)	PA-MPJPE(↓)
linear	*linear*	87.0	51.0
linear	*mixer*	**84.9**	**48.9**
mixer	*mixer*	85.4	50.0

(d) Ablation of Modulator Selection

Table 5. **(a)** Using sampled image features indicated by 2D pose coordinates and our pose tokens as different 2D pose knowledge representation. **(b)** Using embedded raw 2D pose coordinates and tokens from VQ-VAE as source of pose tokens. **(c)** Comparison between other architecture(METRO [29] here) with or without 2D pose tokens and to show the generalizability of 2D pose tokens. **(d)** Using different modulators for different tokens. All of the above results are obtained on 3DPW-TEST [46] and ***without*** fine-tuning on 3DPW-TRAIN split. Best performances are highlighted.

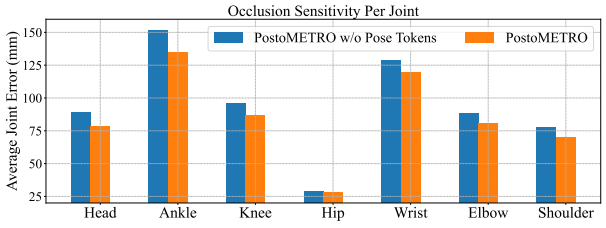


Figure 4. **Per joint occlusion sensitivity analysis** in PostoMETRO with or without joint tokens.

cation across channel dimensions, MLP-Mixer uses shared weights for all channels to do multiplication on spatial dimensions, hence better exploiting the spatial information. However, given the variability of the human body locations in images, using shared weights spatially can spoil image features. This explains why different modulators are needed for different modalities. We also ablate the block number of the MLP-Mixer, please refer to Supp. Mat. for details.

Occlusion Sensitivity. Following [21, 24], we conduct the occlusion sensitivity analysis. Fig. 4 shows the per-joint mean 3D error from the occlusion sensitivity analysis for PostoMETRO with or without pose tokens on 3DPW-TEST. Combining image tokens and pose tokens, our proposed method improves robustness to occlusion in terms of all joints. More occlusion analyses are provided in Supp. Mat.

5. Conclusion

We present PostoMETRO, a novel framework for 3D HPSE tasks incorporating 2D poses and images to enhance robustness to occlusion. Our motivation is to guide model with external 2D pose knowledge to mitigate the training-testing domain gap. Unlike previous methods, PostoMETRO utilizes VQ-VAE to store 2D pose knowledge and represent them as tokens. By combining pose tokens with image tokens, we train transformer-based PostoMETRO in an end-to-end manner. Experiments show that PostoMETRO achieves SOTA performance across various datasets, including scenarios like object-occlusion, person-occlusion, and non-occlusion, without relying on manual design. Extensive ablation studies highlight the benefits of our key design, such as tokenizing 2D poses and leveraging VQ-VAE. We hope this work can contribute to the 3D HPSE community, especially for solving occlusion problems.

6. Acknowledgement

This work is supported by Natural Science Foundation of China under Grant 62271465, Suzhou Basic Research Program under Grant SYG202338, Open Fund Project of Guangdong Academy of Medical Sciences, China (No. YKY-KF202206), and Jiangsu Province Science Foundation for Youths (NO. BK20240464).

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 5

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005. 2

[3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 561–578. Springer, 2016. 2

[4] Zhongang Cai, Wanqi Yin, Ailing Zeng, Chen Wei, Qingping Sun, Wang Yanjun, Hui En Pang, Haiyi Mei, Mingyuan Zhang, Lei Zhang, et al. Smpler-x: Scaling up expressive human pose and shape estimation. *Advances in Neural Information Processing Systems*, 36, 2024. 6

[5] Hanbyel Cho and Junmo Kim. Generative approach for probabilistic human mesh recovery using diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 4183–4188, October 2023. 4

[6] Junhyeong Cho, Kim Youwang, and Tae-Hyun Oh. Cross-attention of disentangled modalities for 3d human mesh recovery with transformers. In *ECCV*, pages 342–359. Springer, 2022. 1, 2, 4, 5, 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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 769–787. Springer, 2020. 2, 6, 7

[8] Hongsuk Choi, Gyeongsik Moon, JoonKyu Park, and Kyoung Mu Lee. Learning to estimate robust 3d human mesh from in-the-wild crowded scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1475–1484, 2022. 2, 3, 5, 6, 7, 8

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2

[11] Zigang Geng, Chunyu Wang, Yixuan Wei, Ze Liu, Houqiang Li, and Han Hu. Human pose as compositional tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 660–671, 2023. 3, 7

[12] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. Humans in 4d: Reconstructing and tracking humans with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14783–14794, 2023. 6

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2

[14] 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, 2013. 2, 4

[15] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 42–52. IEEE, 2021. 2, 6

[16] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131, 2018. 1, 2, 4, 6

[17] Rawal Khirodkar, Shashank Tripathi, and Kris Kitani. Occluded human mesh recovery. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1715–1725, 2022. 7

[18] Jeonghwan Kim, Mi-Gyeong Gwon, Hyunwoo Park, Hyukmin Kwon, Gi-Mun Um, and Wonjun Kim. Sampling is matter: Point-guided 3d human mesh reconstruction. In *CVPR*, pages 12880–12889, 2023. 4, 5

[19] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[20] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020. 6

[21] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11127–11137, 2021. 2, 5, 6, 8

[22] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2252–2261, 2019. 1, 2, 4, 6, 7

[23] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J Black, and Peter V Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6050–6059, 2017. 2, 4

[24] Jiefeng Li, Siyuan Bian, Qi Liu, Jiasheng Tang, Fan Wang, and Cewu Lu. Niki: Neural inverse kinematics with invert-

ible neural networks for 3d human pose and shape estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12933–12942, 2023. 5, 6, 8

[25] 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. 6

[26] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. Cliff: Carrying location information in full frames into human pose and shape estimation. In *European Conference on Computer Vision*, pages 590–606. Springer, 2022. 2, 6

[27] Jing Lin, Ailing Zeng, Haoqian Wang, Lei Zhang, and Yu Li. One-stage 3d whole-body mesh recovery with component aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21159–21168, 2023. 6

[28] Kevin Lin, Chung-Ching Lin, Lin Liang, Zicheng Liu, and Lijuan Wang. Mpt: Mesh pre-training with transformers for human pose and mesh reconstruction. *arXiv preprint arXiv:2211.13357*, 2022. 3, 5, 6

[29] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, pages 1954–1963, 2021. 1, 2, 4, 5, 6, 7, 8

[30] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, pages 12939–12948, 2021. 5, 6

[31] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3, 5

[32] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 1, 2

[33] Xiaoxuan Ma, Jiajun Su, Chunyu Wang, Wentao Zhu, and Yizhou Wang. 3d human mesh estimation from virtual markers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 534–543, June 2023. 5, 6

[34] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 3

[35] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3d pose estimation from monocular rgb. In *2018 International Conference on 3D Vision (3DV)*, pages 120–130. IEEE, 2018. 2, 4

[36] Gyeongsik Moon, Hongsuk Choi, Sanghyuk Chun, Jiyoung Lee, and Sangdoo Yun. Three recipes for better 3d pseudo-gts of 3d human mesh estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2754–2763, 2023. 2

[37] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Neuralannot: Neural annotator for 3d human mesh training sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2299–2307, 2022. 2

[38] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. 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*, pages 10975–10985, 2019. 2

[39] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. 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*, pages 10975–10985, 2019. 2

[40] Zhongwei Qiu, Qiansheng Yang, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, Chang Xu, Dongmei Fu, and Jingdong Wang. Psvt: End-to-end multi-person 3d pose and shape estimation with progressive video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21254–21263, 2023. 3, 6

[41] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. 3

[42] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European conference on computer vision (ECCV)*, pages 704–720, 2018. 4

[43] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei. Monocular, one-stage, regression of multiple 3d people. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11179–11188, 2021. 2, 5, 6, 7, 8

[44] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021. 3

[45] Gul Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3d human body shapes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 20–36, 2018. 2

[46] Timo Von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European conference on computer vision (ECCV)*, pages 601–617, 2018. 2, 5, 6, 8

[47] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui

Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 1, 2

[48] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6184–6193, 2020. 2

[49] Sen Yang, Wen Heng, Gang Liu, GUOZHONG LUO, Wankou Yang, and Gang YU. Capturing the motion of every joint: 3d human pose and shape estimation with independent tokens. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023. 6

[50] Chun-Han Yao, Jimei Yang, Duygu Ceylan, Yi Zhou, Yang Zhou, and Ming-Hsuan Yang. Learning visibility for robust dense human body estimation. In *European Conference on Computer Vision*, pages 412–428. Springer, 2022. 2, 6

[51] 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/CVF International Conference on Computer Vision*, pages 11446–11456, 2021. 1, 2, 7

[52] Song-Hai Zhang, Ruilong Li, Xin Dong, Paul Rosin, Zixi Cai, Xi Han, Dingcheng Yang, Haozhi Huang, and Shi-Min Hu. Pose2seg: Detection free human instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 889–898, 2019. 5

[53] Tianshu Zhang, Buzhen Huang, and Yangang Wang. Object-occluded human shape and pose estimation from a single color image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7376–7385, 2020. 2, 5, 6, 8