MPT: Mesh Pre-Training with Transformers for Human Pose and Mesh Reconstruction

Kevin Lin, Chung-Ching Lin, Lin Liang, Zicheng Liu, Lijuan Wang
Microsoft
{keli, chungching.lin, lliang, zliu, lijuanw}@microsoft.com

Abstract

Traditional methods of reconstructing 3D human pose and mesh from single images rely on paired image-mesh datasets, which can be difficult and expensive to obtain. Due to this limitation, model scalability is constrained as well as reconstruction performance. Towards addressing the challenge, we introduce Mesh Pre-Training (MPT), an effective pre-training strategy that leverages large amounts of MoCap data to effectively perform pre-training at scale. We introduce the use of MoCap-generated heatmaps as input representations to the mesh regression transformer and propose a Masked Heatmap Modeling approach for improving pre-training performance. This study demonstrates that pre-training using the proposed MPT allows our models to perform effective inference without requiring fine-tuning. We further show that fine-tuning the pre-trained MPT model considerably improves the accuracy of human mesh reconstruction from single images. Experimental results show that MPT outperforms previous state-of-the-art methods on Human3.6M and 3DPW datasets. As a further application, we benchmark and study MPT on the task of 3D hand reconstruction, showing that our generic pre-training scheme generalizes well to hand pose estimation and achieves promising reconstruction performance.

1. Introduction

3D human pose and mesh reconstruction from a single image is a challenging task in computer vision [11, 13, 23, 25–27, 29, 40], which involves estimating the 3D coordinates of human body joints and mesh vertices from a 2D image. The task has various applications in areas such as human motion analysis and human-centric event understanding. Recent advances in transformer models [11, 30, 31] have shown remarkable success in this task. Most of the models are, however, trained in a supervised manner using paired image-mesh datasets, which are costly to acquire in practice. This has limited model scalability and performance, and also the development of 3D pose reconstruction.

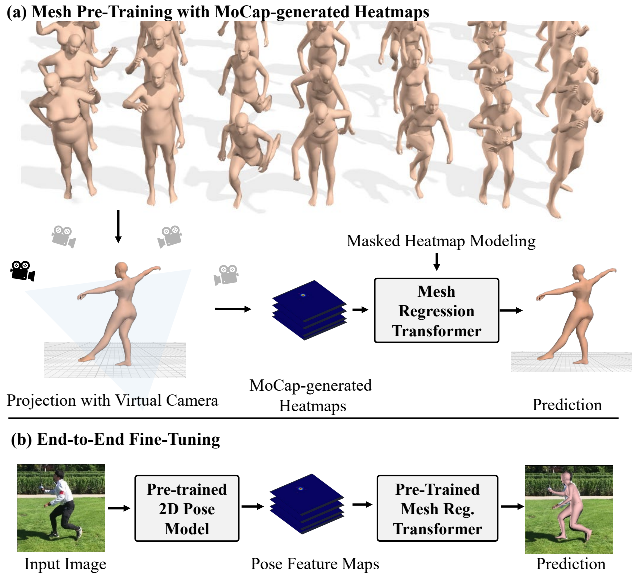

(a) Mesh Pre-Training with MoCap-generated Heatmaps
Masked Heatmap Modeling
Mesh Regression Transformer
Projection with Virtual Camera
MoCap-generated Heatmaps
Prediction
(b) End-to-End Fine-Tuning
Pre-trained 2D Pose Model
Pre-Trained Mesh Reg. Transformer
Input Image
Pose Feature Maps
Prediction

Figure 1. Summary of our pretrain-finetune strategy. (a) In our proposed Mesh Pre-Training, we pre-train mesh regression transformer using MoCap-generated heatmaps to learn human pose and shape knowledge. (b) After pre-training, we use an off-the-shelf 2D pose model to extract pose feature maps, which are then fed to the mesh regression transformer. Through end-to-end fine-tuning, our model learns to reconstruct 3D human pose and mesh from the input image.

To address the challenge, prior works [13, 14, 37, 45, 65] attempted to use 3D mesh data, such as motion capture (MoCap) data [1, 35], for training a 2D-to-3D lifting model that projects the 2D joint coordinates into 3D space. Such 3D mesh data has proven to be beneficial in learning detailed skeleton articulations. Despite providing accurate 3D joint coordinates along with the body meshes, MoCap data generally lack corresponding RGB images. The majority of these approaches, therefore, do not involve the use of images in learning and could be prone to depth ambiguity.

In this paper, we present an effective pre-training method called Mesh Pre-Training (MPT) that leverages large amounts of MoCap data and learns with MoCap-generated heatmaps. Instead of directly performing a 2D-to-3D lifting task, as shown in Figure 1(a), we propose to pre-train the mesh regression transformer by using MoCap-generated heatmaps, which are synthesized from the 2D joint coordinates obtained from virtual cameras for each 3D mesh sample. These heatmaps serve as input representations for the mesh regression transformer during pre-training. To facilitate self-attention learning with such input representations, we propose a Masked Heatmap Modeling (MHM) approach to improve pre-training performance.

After pre-training, as shown in Figure 1(b), we use an off-the-shelf 2D human pose estimation model to extract pose feature maps, which are then fed to the mesh regression transformer for human mesh reconstruction. We then fine-tune both the 2D human pose estimation model and the mesh regression transformer in an end-to-end manner. As we will show in the experiments, our model learns to extract pose feature maps with a more general focus on multiple body joints. Accordingly, the pose feature maps together with the pre-trained mesh regression transformer contribute to the reconstruction of high-fidelity human meshes.

Our MPT model is pre-trained on a large-scale MoCap dataset consisting of 2 million human meshes. After pre-training, we fine-tune and evaluate the model on target datasets for single-image 3D human pose and mesh reconstruction. Experimental results demonstrate that our proposed MPT outperforms the previous state-of-the-art approaches on multiple public benchmarks, including Human3.6M and 3DPW datasets. In addition, our experimental results suggests that the proposed MPT enables inference capability without the need of fine-tuning.

Furthermore, we demonstrate the versatility of our approach by applying MPT to the task of single-image 3D hand reconstruction, achieving state-of-the-art results on the FreiHAND dataset. The proposed MPT approach provides a promising direction for scaling up pre-training for single-image 3D human pose and mesh reconstruction without the need for paired image-mesh datasets.

In summary, we make the following contributions.

- We present a simple yet effective pre-training method, called Mesh Pre-Training (MPT), for the 3D human pose and mesh reconstruction from single images.
- We introduce the use of MoCap-generated heatmaps with Masked Heatmap Modeling (MHM) for pre-training mesh regression transformer.
- The proposed method outperforms previous state-of-the-art methods on multiple benchmarks including Human3.6M, 3DPW, and FreiHAND.

2. Related Works

Single-image human pose and mesh reconstruction: Prior works can be clustered into two categories: parametric and non-parametric approaches. Parametric approaches [17,23,24,26,28,45,56] typically adopt the SMPL model [34] and regress SMPL parameters to generate human meshes. While SMPL model has shown great success and is convenient and robust to pose variations, it is challenging to estimate accurate SMPL parameters from a single image. Recent studies [18, 41, 44, 56, 63] have been focusing on various auxiliary supervisions such as improving 2D re-projection [29] or extending it to video-based methods [12, 24, 59] to improve the estimation of SMPL parameters.

Different from adopting SMPL as the regression target, non-parametric approaches [11, 13, 27, 30, 31, 40] aim to predict the 3D coordinates of body joints and mesh vertices directly from the input image. Researchers have explored graph convolutional neural networks [13, 27] as well as transformer architecture [11, 30, 31] which is effective in modeling vertex-vertex and vertex-joint interactions for improving the reconstruction performance.

Our work draws inspiration from Pose2Mesh [13], a related study that employs a multi-stage 2D-to-3D lifting pipeline. Pose2Mesh begins by detecting 2D body joint locations and then elevating these coordinates into 3D space. It subsequently reconstructs a 3D mesh based on the lifted 3D joint coordinates. In comparison, we propose to use MoCap-generated heatmaps to pre-train the mesh regression transformer. Our design enables end-to-end training and considerably improves the performance of human mesh reconstruction.

Synthetic training data generation: Previous studies have explored 3D graphics rendering [4, 6, 47, 57] to generate large amount of image-mesh pairs. However, models trained on synthetic data may not perform well on real images due to the domain gap between synthetic and real data [16, 53, 54]. Recent studies [22, 29, 39] have proposed training a human mesh annotator to generate 3D pseudo labels on real images. Unlike existing works that collect image-mesh pairs, our method uses MoCap-generated heatmaps to pre-train models using only 3D mesh data. Our method does not rely on the photorealism of the synthetic data generation, and provides an effective way to leverage 3D mesh datasets that lack associated RGB images.

Human motion capture (MoCap) datasets: There are many optical marker-based motion capture data available [2, 36, 42, 55], including CMU [1], and AMASS [35]. MoCap data records a variety of human body movements, which is useful for human motion generation and analysis [24, 46, 51, 52]. Since MoCap data is usually captured by the infrared (IR) sensors and performers have to wear special cloth and markers, there are usually no RGB images

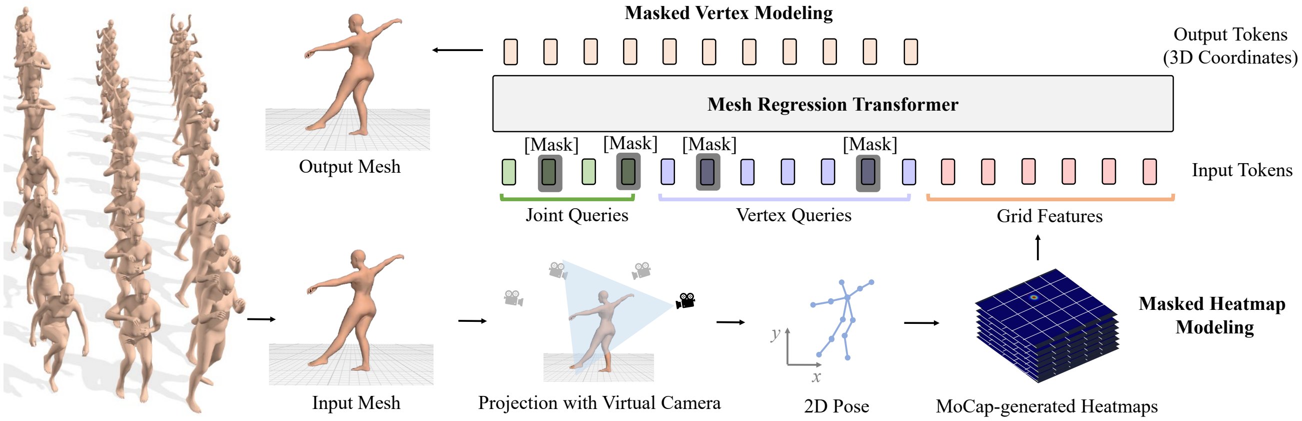


Figure 2. Overview of the proposed Mesh Pre-Training (MPT). Given a human mesh which is sampled from MoCap data, we project its 3D human pose to 2D by randomly selecting a virtual camera. We then synthesize the heatmaps to represent the projected 2D human pose. Our mesh regression transformer takes three types of input tokens, including joint queries, vertex queries, and MoCap-generated heatmaps. We then pre-train the transformer to predict the 3D coordinates of body joints and mesh vertices given the input tokens. During pre-training, we perform Masked Heatmap Modeling and Masked Vertex Modeling to improve the robustness of the mesh regression transformer.

available. To deal with the problem, prior works [4, 47, 57] proposed to leverage 3D graphics engines to synthesize RGB images for the collection of image-mesh pairs. More recently, researchers have proposed to train a motion discriminator [24] to help improve video-based human mesh reconstruction. Different from prior works, we use MoCap datasets to pre-train our mesh regression transformer which improves the accuracy of 3D pose and mesh reconstruction from a single image.

3. Method

Our approach uses a two-stage training scheme that consists of (*i*) mesh pre-training, (*ii*) end-to-end fine-tuning. Figure 1 illustrates our approach using an example. First, in the pre-training stage (Figure 1(a)), we pre-train the mesh regression transformer to learn human mesh reconstruction by using MoCap-generated heatmaps. These heatmaps are synthesized from 2D joint coordinates which are obtained by projecting 3D joints with a randomly selected virtual camera. Second, Figure 1(b) illustrates our framework at the fine-tuning stage. We use an off-the-shelf pre-trained 2D pose estimation model to extract pose feature maps, which are then fed into the mesh regression transformer for human mesh reconstruction. We then perform end-to-end fine-tuning on the target dataset.

3.1. Pre-Training and MoCap-generated Heatmaps

As shown in Figure 2, our mesh regression transformer takes three types of tokens as inputs, including joint queries, vertex queries, and MoCap-generated heatmaps. The mesh regression transformer is asked to directly regress the 3D coordinates of body joints and mesh vertices from MoCap-generated heatmaps.

We pre-train our mesh regression transformer using a large scale MoCap dataset (*e.g.*, AMASS dataset [35]). Unlike existing works that rely on image-mesh pairs, the proposed Mesh Pre-Training (MPT) is conducted by using 3D mesh data without RGB images. As AMASS dataset consists of sequences of SMPL human meshes, we sparsely sample the human meshes from each motion capture sequence, and then obtain the corresponding 3D human poses using SMPL regressor. This results in a total of 2 million meshes. Note that all the meshes are normalized and centered at the origin. We pre-define 4 virtual cameras above the head positions of the human meshes. For each mesh and each virtual camera, we obtain the projected 2D joint coordinates which are then used to synthesize the joint heatmaps. In this way, we generate 8 million heatmap-mesh pairs which are used in pre-training. More details on the heatmap generation are discussed next.

3.1.1 Synthesizing Heatmaps

Given a mesh, we first project the 3D pose to 2D by randomly selecting a virtual camera. We then represent the 2D pose in the form of heatmaps [10, 48, 60] (also called confidence maps [7]). To be specific, we generate a set of heatmaps $\mathbf{S}$ based on the projected 2D pose. The set $\mathbf{S} = (\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_K)$ has K heatmaps, one per joint, where K is the total number of body joints, $\mathbf{S}_j \in \mathbb{R}^{w \times h}$ for $j \in \{1, 2, \dots, K\}$. We set $K = 17$ in our experiments following the body joint definition in COCO dataset [32]. Each heatmap depicts the 2D position of a specified body joint. Let x_j be the 2D coordinate of the body joint j. The value at location $\mathbf{p}$ in the heatmap $\mathbf{S}_j$ is

$$\mathbf{S}_j(\mathbf{p}) = \exp\left(-\frac{||\mathbf{p} - x_j||_2^2}{\sigma^2}\right), \quad (1)$$

where $\sigma = 3$ following the literature [48].

In order to input the heatmaps to the transformer model, we concatenate the heatmaps $\mathbf{S}$ along the channel dimension to form a 3D tensor, which is of size $(W \times H \times C)$, and $W = H = 224$. Similar to ViT [15], we split the tensor into non-overlapping patches using a patch partition module [15, 33]. Each patch is then treated as an input token (or grid feature [21]) to the transformer model. In our implementation, we use a patch size of (8×8). That is, the feature dimension of each patch is $(8 \times 8 \times C)$. Finally, we apply a multi-layer perceptron (MLP) layer to make the dimension of grid features consistent with the hidden size of the transformer model.

3.1.2 Masked Heatmap Modeling

To facilitate the learning of transformer self-attention with MoCap-generated heatmaps, during pre-training, we randomly mask some of the joints in the MoCap-generated heatmaps, and ask the mesh regression transformer to predict all the 3D body joints and mesh vertices. Unlike existing works [30, 31] that only used Masked Vertex Modeling to randomly mask out some of the query tokens, we perform masking on the heatmaps. Our masking mechanism is in spirit similar to simulating the real heatmaps obtained from the off-the-shelf 2D pose estimation model, where some joints might be missing. In addition to the masking, we also apply data augmentation to the heatmaps, including adding Gaussian noise and joint coordinate jittering.

3.1.3 Pre-Training Objective

Our pre-training objective is a regression task conditioned on the MoCap-generated heatmaps. To regress the mesh vertices, following the literature [11, 27, 30, 31], we use L_1 loss to minimize the differences between the predicted vertices V_{3D} and the ground truth vertices $\bar{V}_{3D}$:

$$\mathcal{L}_V = \frac{1}{M} \sum_{i=1}^{M} \left|\left| V_{3D} - \bar{V}_{3D} \right|\right|_1 , \tag{2}$$

where $\bar{V}_{3D} \in \mathbb{R}^{M \times 3}$, and M is the total number of vertices.

In addition, we minimize the differences between the predicted joints J_{3D} and the ground truth joints $\bar{J}_{3D}$:

$$\mathcal{L}_J = \frac{1}{K} \sum_{i=1}^{K} \left|\left| J_{3D} - \bar{J}_{3D} \right|\right|_1 , \tag{3}$$

where K is the total number of 3D joints.

Note that the 3D joints can be regressed from the mesh vertices using a pre-defined matrix G [11, 13, 23, 26, 27, 29]. We also apply supervision on the regressed 3D joints:

$$\mathcal{L}_J^{reg} = \frac{1}{K} \sum_{i=1}^{K} \left|\left| J_{3D}^{reg} - \bar{J}_{3D} \right|\right|_1 , \tag{4}$$

where $J_{3D}^{reg} = GV_{3D}$, and $G \in \mathbb{R}^{K \times M}$.

Following the common practice [11, 23, 26, 27, 29–31], we also employ the 2D re-projection loss. Given the predicted 3D joints, we project the predicted 3D joints to 2D using the estimated camera parameters. We then minimize the errors between the projected 2D joints J_{2D} and the ground truth 2D joints $\bar{J}_{2D}$:

$$\mathcal{L}_J^{proj} = \frac{1}{K} \sum_{i=1}^{K} \left|\left| J_{2D} - \bar{J}_{2D} \right|\right|_1 . \tag{5}$$

Finally, our pre-training objective can be written as

$$\mathcal{L}_{\text{pre-train}} = \mathcal{L}_V + \mathcal{L}_J + \mathcal{L}_J^{reg} + \mathcal{L}_J^{proj} . \tag{6}$$

3.2. Testing without Fine-tuning

After pre-training, the pre-trained MPT model can be directly applied to any real images for human mesh reconstruction. Given an input image, we use an off-the-shelf 2D pose estimation model to extract the pose feature maps. A 2D pose estimation model typically predicts a set of heatmaps (one heatmap per joint), followed by a series of post-processing to obtain the 2D joint coordinates. We remove those post-processing operations. The remaining network is used as our backbone network to extract pose feature maps from a given image.

We then input the extracted feature maps to the mesh regression transformer for 3D human pose and mesh reconstruction. In this way, our pre-trained MPT model is capable of human mesh reconstruction in a plug-and-play fashion.

It is worth noting that there are many well-performing pre-trained 2D pose estimation models [7, 9, 10, 43, 48, 60, 61] available, and we use a recently developed one (*e.g.*, HigherHRNet [10]) in our main experiments. In addition, we will show that our proposed MPT is not sensitive to the choices of 2D pose estimation model.

3.3. End-to-End Fine-Tuning

After pre-training, the pre-trained MPT model can be fine-tuned to the target dataset for more accurate reconstruction. Given the image-mesh pairs from the target dataset, we fine-tune both the backbone network and mesh regression transformer in an end-to-end manner. We use the same loss function as that in our pre-training stage.

Similar to the existing studies [11, 23, 26, 27, 29–31, 40], we fine-tune our model using a mixture of 2D and 3D datasets. We calculate the losses as long as the ground truths are available.

3.4. Implementation Details

Our mesh regression transformer is in spirit similar to METRO [30]. It consists of multiple transformer layers to

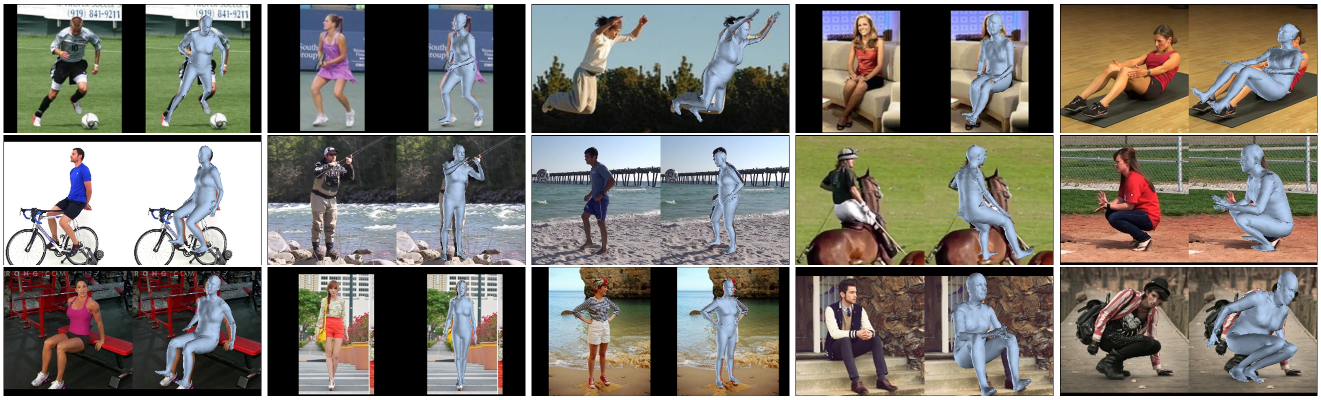


Figure 3. Testing without fine-tuning. After pre-training, we test our MPT model on real images directly. MPT generates reasonable human meshes on real images without any fine-tuning. This demonstrates the generalizability of our MPT model.

regress the 3D coordinates of mesh vertices and body joints. An important difference of our model is that we take pose feature maps as inputs to the transformer network. We empirically observed the design is effective to bridge the 2D pose estimation network and the pre-trained mesh regression transformer.

4. Experimental Results

In this section, we first discuss the datasets we used in pre-training and fine-tuning. We then present the performance comparison with existing state-of-the-arts on public benchmarks. Finally, we provide a detailed ablation study to verify the effectiveness of the proposed training scheme.

4.1. Datasets and Evaluation Metrics

We conduct pre-training on AMASS collection [35], which consists of 24 MoCap datasets. AMASS provides a unified human pose and mesh representations based on SMPL [34]. Each sequence records a motion movement of a subject. In total, there are 500 subjects with $17,916$ motions. The total length of all the sequences is about $3,772$ minutes. In total, there are about $25,088,088$ frames, and each frame has a human mesh. To avoid the redundancy between the neighbor frames, we sparsely sample 2 million meshes from AMASS. After projection using 4 virtual cameras, we obtain 8 million heatmap-mesh pairs for pre-training.

We fine-tune our model using a mixture of 2D and 3D datasets, including Human3.6M [20], MuCo-3DHP [38], UP-3D [28], COCO [32], and MPII [3]. Note that these datasets are commonly used in literature [11, 13, 30, 31]. After that, we evaluate our model on Human3.6M using P2 protocol [23, 26]. When conducting experiments on 3DPW [58], we follow the prior works [11, 24, 30, 31] and fine-tune with 3DPW training data. We then evaluate the results on 3DPW test set.

Following literature [11, 23, 26, 27, 29], we use three standard metrics for evaluation, including Mean Per Joint Position Error (MPJPE) [20], Procrustes Analysis with MPJPE (PA-MPJPE) [66], and Mean Per Vertex Error (MPVE) [45]. The unit of the metrics is millimeter (mm).

	3DPW			Human3.6M	
Method	MPVE ↓	MPJPE ↓	PA-MPJPE ↓	MPJPE ↓	PA-MPJPE ↓
HMR [23]	–	–	81.3	88.0	56.8
GraphCMR [27]	–	–	70.2	–	50.1
SPIN [26]	116.4	96.9	59.2	62.5	41.1
Pose2Mesh [13]	–	89.2	58.9	64.9	47.0
I2LMeshNet [40]	–	93.2	57.7	55.7	41.1
PyMAF [64]	110.1	92.8	58.9	57.7	40.5
ROMP [49]	93.4	76.7	47.3	–	–
VIBE [24]	99.1	82.0	51.9	65.6	41.4
METRO [30]	88.2	77.1	47.9	54.0	36.7
THUNDER [62]	88.0	74.8	51.5	48.0	34.9
PARE [25]	88.6	74.5	46.5	–	–
Graphormer [31]	87.7	74.7	45.6	51.2	34.5
FastMETRO [11]	84.1	73.5	44.6	52.2	33.7
CLIFF [29]	81.2	69.0	43.0	47.1	32.7
MPT (Ours)	**79.4**	**65.9**	**42.8**	**45.3**	**31.7**

Table 1. Performance comparison with the previous state-of-the-art methods on 3DPW and Human3.6M datasets.

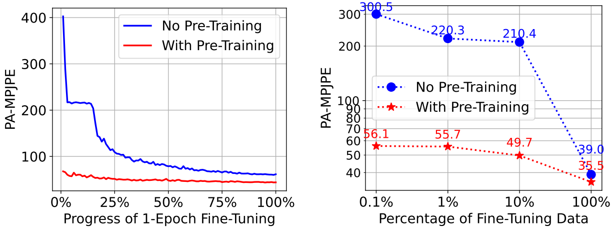
4.2. Main Results

We compare our method with the existing state-of-the-art approaches on Human3.6M [20] and 3DPW [58] datasets. We present our pretrain-then-finetune results in Table 1. Our method outperforms the previous works on both datasets, including the recent transformer-based methods [11, 30, 31, 62].

It is worth noting that, CLIFF [29] was the state-of-the-art approach on the two datasets. CLIFF additionally leverages the camera focal length and bounding box information to calculate 2D re-projection loss on the non-cropped input image. In contrast, we do not have such post-processing, and still achieve better results, especially on MPJPE.

Method	MPT	FT	MPJPE ↓	PA-MPJPE ↓
HMR [23]	✗	Mixed Datasets	88.0	56.8
METRO [30]	✗	Mixed Datasets	54.0	36.7
MPT	✗	Human3.6M	59.1	39.2
MPT	✗	Mixed Datasets	46.6	32.4
MPT (Test w/o fine-tune)	✓	✗	88.9	58.4
MPT	✓	Human3.6M	53.3	35.5
MPT	✓	Mixed Datasets	45.3	31.7

Table 2. Pre-training analysis. We conduct training with different configurations, and then evaluate results on Human3.6M validation set. MPT: Mesh Pre-Training. FT: Fine-Tuning.



(a) Learning Convergence

(b) Fine-Tune with Less Data

Figure 4. Fine-tuning behavior on Human3.6M. **(a)** We conduct a 1-epoch fine-tuning and report PA-MPJPE for each fine-tuning step. **(b)** We use a percentage of Human3.6M data for fine-tuning and report PA-MPJPE.

4.3. Analysis

Effectiveness of Mesh Pre-Training: We study whether our pre-training is useful for performance improvements. We conduct experiments on Human3.6M, and Table 2 shows the comparison with different training configurations including with or without mesh pre-training, and different datasets for fine-tuning. Because our model architecture is similar to METRO [30], we include it as the reference. We also include the well-known HMR [23] as reference.

In Table 2, our MPT improves the performance across different training configurations considered. When fine-tuning our pre-trained model using Human3.6M only, as shown in the sixth row, MPT achieves 35.5 PA-MPJPE, which is better than 39.2 PA-MPJPE of our non-pretrain model. We observe similar findings when using multiple datasets for fine-tuning. Our MPT achieves 31.5 PA-MPJPE, and is better than 32.4 PA-MPJPE of our non-pretrain model.

Testing without Fine-Tuning: Given the pre-trained MPT model, we directly evaluate MPT model on real images without any fine-tuning. As shown in the fifth row of Table 2, we obtain a performance of 58.4 PA-MPJPE. Although it lags behind the supervised fully fine-tuned models, the result is comparable to the well-known HMR [23]. Figure 3 shows the qualitative results. We see that MPT is capable of generating human meshes on real images without the need of fine-tuning.

Learning Convergence: We study the impact of MPT during fine-tuning. We perform a 1-epoch fine-tuning on Human3.6M, and report results for each fine-tuning step. Figure 4a shows that, with our pre-training, the fine-tuning converges better than that without pre-training.

Fine-Tuning with Less Data: We select 0.1%, 1%, and 10% of Human3.6M training data for fine-tuning, respectively. Figure 4b shows that our pre-training helps improve the learning performance when less data is used during fine-tuning. For example, our pre-trained model with 0.1% fine-tune data achieves 56.1 PA-MPJPE, which is much better than 210.4 PA-MPJPE of our non-pretrained model with 10% fine-tune data.

Pre-Training with MoCap-generated Heatmaps: Since we use MoCap-generated heatmaps during our pre-training, one important question is that what if we directly input 2D joint coordinates to the mesh regression transformer. Table 3a shows the comparison of the pre-training performance (*i.e.,* testing without fine-tuning) on Human3.6M. We observe that the use of MoCap-generated heatmaps effectively improves the pre-training performance.

Fine-Tuning with Pose Feature Maps: During the fine-tuning stage, our mesh regression transformer takes pose feature maps as inputs. One may wonder what if we use image feature maps instead, as discussed in Mesh Graphormer [31]. To investigate the impact of this design choice, we conducted an ablation study on Human3.6M, as shown in Table 3b. The results indicate that using pose feature maps during the fine-tuning stage leads to better performance across all considered metrics.

Analysis of Pose Feature Maps: In Figure 6, we present a visualization of the pose feature maps during fine-tuning. The top row of Figure 6 shows that when the backbone is frozen during fine-tuning, the pose feature maps behave like traditional heatmaps, capturing a single body joint location per map. We can see that the reconstructed mesh is not correct due to the side view angle and self occlusions. In contrast, as shown in the bottom row of Figure 6, when we fine-tune both the backbone and the mesh regression transformer, our model learns to extract pose feature maps where each map captures information on multiple body joints, resulting in improved reconstruction and more accurate pose estimation. For example, in order to predict the location of the left ear, our model pays attention to not only the keypoints on the face, but also the body joints on the arms and legs. In Table 4, we provide a quantitative comparison between freezing and unfreezing backbone during fine-tuning. The results suggest end-to-end fine-tuning improves the reconstruction performance.

Effectiveness of Masked Heatmap Modeling: Table 3c shows an ablation study of the proposed Masked Heatmap Modeling (MHM), tested on Human3.6M. We observe MHM improves the reconstruction performance by a large margin, especially on PA-MPJPE metric.

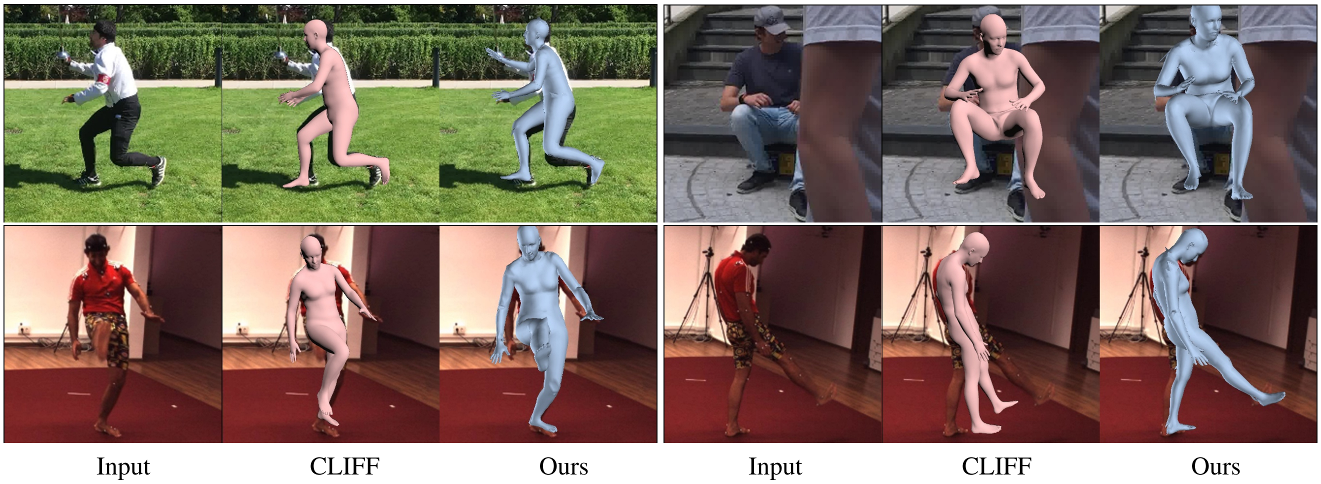


Figure 5. Qualitative comparison. For each example, we show the results from CLIFF [29] and our proposed MPT. Both CLIFF and MPT generate good quality human meshes, but MPT has more favorable body pose. Tow row: 3DPW. Bottom row: Human3.6M.

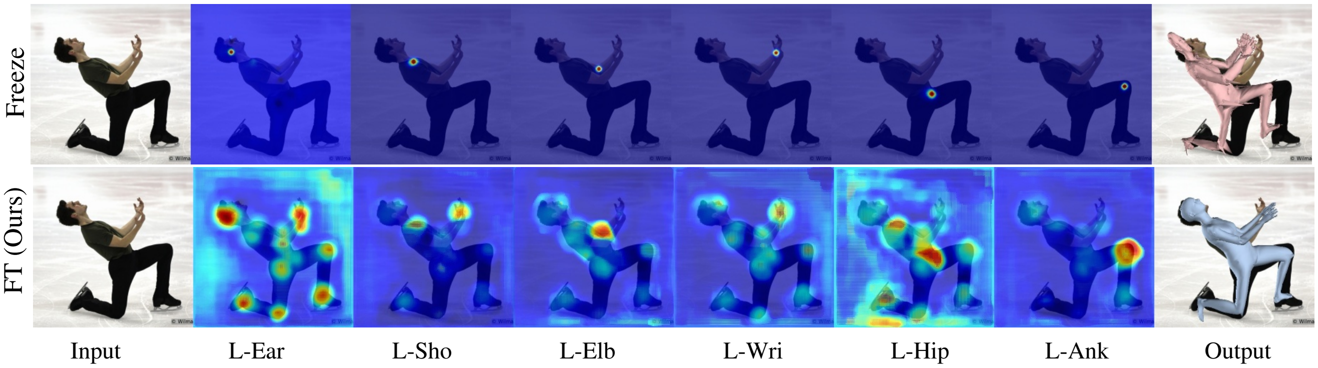


Figure 6. Visualization of pose feature maps. In the top row, we freeze the backbone and fine-tune only the mesh regression transformer. However, we observe that the pose feature maps behave like conventional heatmaps, detecting one joint per map and resulting in incorrect reconstruction. In the bottom row, we unfreeze the backbone and perform end-to-end fine-tuning. Our model learns to extract pose feature maps where each map captures information on multiple body joints, leading to improved reconstruction and more accurate pose estimation. More visualizations are provided in the supplementary material.

Pre-Training	MPJPE ↓	PA-MPJPE ↓
2D Coordinates	139.3	88.9
MoCap-gen. Heatmaps	88.9	58.4

(a) Comparison of Different Pre-Training: MoCap-generated Heatmaps vs. 2D Coordinates

Fine-Tuning	MPT	FT	MPJPE ↓	PA-MPJPE ↓
Image Features	✗	✓	78.0	47.0
Pose FeatMaps	✗	✓	58.7	39.7
Pose FeatMaps	✓	✓	53.3	35.5

(b) Different Inputs During Fine-Tuning

Method	MPJPE ↓	PA-MPJPE ↓
MPT w/o MHM	97.7	64.4
MPT w/ MHM	88.9	58.4

(c) Masked Heatmap Modeling

Table 3. **(a)** We study pre-training using different input representations to the mesh regression transformer. We evaluate pre-training performance (*i.e.,* testing without fine-tuning) on Human3.6M. **(b)** During fine-tuning, we study the effect of different input representations to the mesh regression transformer, including image features and pose feature maps. We conduct fine-tuning and evaluation on Human3.6M. **(c)** Ablation study of Masked Heatmap Modeling. We evaluate our pre-trained MPT on Human3.6M without any fine-tuning.

Backbone	FT	MPJPE ↓	PA-MPJPE ↓
Freeze	✓	78.1	53.5
Unfreeze	✓	53.3	35.5

Table 4. Comparison between freeze and unfreeze backbone during fine-tuning. We conducted fine-tuning on Human3.6M.

Percentage of PT Data	20%	40%	60%	80%	100%
PA-MPJPE ↓	63.4	62.4	59.9	59.4	58.4

Table 5. Analysis of pre-training data size. We conduct pre-training using different percentage of data, and evaluate the pre-trained MPT on Human3.6M without fine-tuning.

Pre-Training Data Size: Since we use 2 million meshes for pre-training, an interesting question is how much data should we use for pre-training. To answer the question, we conduct pre-training using different percentage of the 2 million meshes. We then evaluate our MPT model directly on Human3.6M validation set without fine-tuning. Table 5

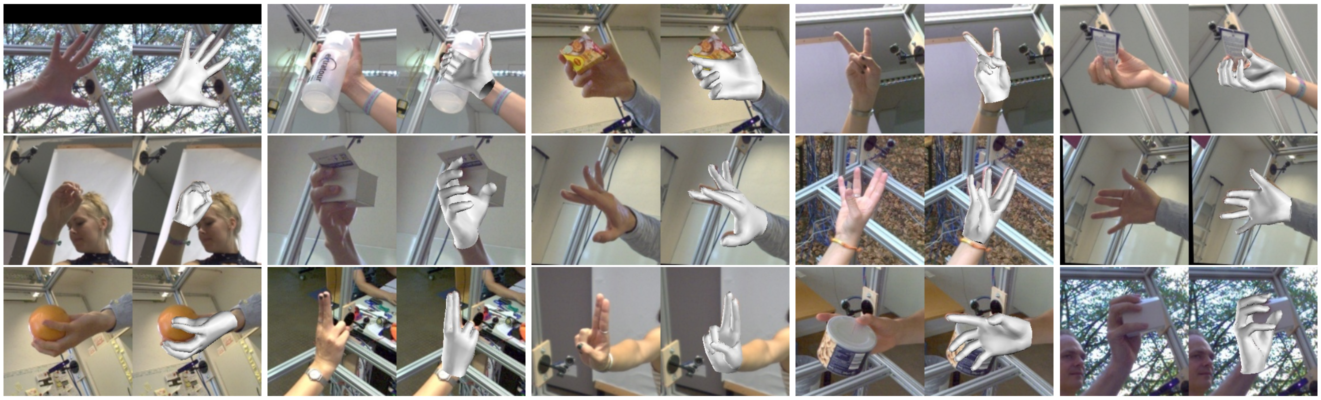


Figure 7. Qualitative results of MPT applying to 3D hand reconstruction on FreiHAND dataset.

Method	PA-MPVPE ↓	PA-MPJPE ↓	F@5 mm ↑	F@15 mm ↑
Hasson et al [19]	13.2	–	0.436	0.908
Boukhayma et al. [5]	13.0	–	0.435	0.898
FreiHAND [67]	10.7	–	0.529	0.935
Pose2Mesh [13]	7.8	7.7	0.674	0.969
I2LMeshNet [40]	7.6	7.4	0.681	0.973
METRO [30]	6.8	6.7	0.717	0.981
Tang *et al.* [50]	6.7	6.7	0.724	0.981
FastMETRO [11]	–	6.5	–	0.982
Graphormer [31]	6.0	5.9	0.764	0.986
MobRecon [8]	5.7	5.8	0.784	0.986
MPT (Ours)	**5.4**	**5.6**	**0.789**	**0.988**

Table 6. Performance comparison with the previous state-of-the-art methods on FreiHAND dataset.

Number of Views	1	2	4
PA-MPJPE ↓	61.7	61.4	58.4

Table 7. Ablation study of numbers of virtual camera views. We evaluate the pre-trained MPT on Human3.6M without fine-tuning.

Backbone	MPJPE ↓	PA-MPJPE ↓
SimpleBaseline [61]	45.8	32.8
HigherHRNet [10]	46.6	32.4

Table 8. Comparison of different human pose networks. We conduct end-to-end fine-tuning on mixed 2D/3D datasets, and evaluate on Human3.6M. No pre-training in this ablation study.

shows that increasing the number of meshes for pre-training can improve the performance on Human3.6M. The results suggest that we could scale-up our pre-training data for further performance improvements, and we leave it as future work.

Number of Virtual Camera Views: We also study the effect of different number of virtual camera views. Table 7 shows that adding more camera views can improve the performance. The results suggest increasing the training data diversity is beneficial to our pre-training.

Backbone Analysis: As we use HigherHRNet as our backbone to extract feature maps, one may wonder what if we use other backbones. In Table 8, we replace HigherHRNet with an early baseline called SimpleBaseline [61]. Two backbones have quite different mAPs on COCO keypoint dataset. But after adding the backbone to MPT for training, both MPT variants give similar results, which suggests MPT is not very sensitive to the backbone choices.

Qualitative Results: Figure 5 shows the qualitative results of MPT compared with CLIFF [29] on Human3.6M and 3DPW datasets. While both methods can generate good quality human meshes, MPT has more favorable body poses when there are self-occlusions in the images.

Applying to 3D Hand Reconstruction: MPT is a generic pre-training scheme for human mesh regression task. We demonstrate the flexibility of MPT on 3D hand reconstruction, and we conduct the experiments on FreiHAND [67] dataset. Since there is less MoCap data for 3D hands, we use a recently proposed synthetic dataset called Complement [8] for pre-training. Table 6 shows the performance comparison with previous works. MPT outperforms the previous state-of-the-art methods, including MobRecon [8] which also uses Complement as training data. Figure 7 shows the qualitative examples of our hand reconstruction.

5. Conclusion

We introduced Mesh Pre-Training (MPT), a simple yet effective pre-training strategy that leverages large-scale MoCap data to pretrain the mesh regression transformer for 3D human pose and mesh reconstruction from a single image. We introduce the use of MoCap-generated heatmaps with Masked Heatmap Modeling for pre-training the mesh regression transformer. Experimental results show that our method outperforms the previous state-of-the-art methods on Human3.6M, 3DPW, and FreiHAND datasets. We further show that MPT can be directly applied to real images without fine-tuning on any image-mesh pairs.

References

[1] Carnegie Mellon University Motion Capture Database. http://mocap.cs.cmu.edu/. 1, 2

[2] Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *CVPR*, 2015. 2

[3] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 5

[4] Fabien Baradel, Thibault Groueix, Philippe Weinzaepfel, Romain Brégier, Yannis Kalantidis, and Grégory Rogez. Leveraging mocap data for human mesh recovery. In *3DV*, 2021. 2, 3

[5] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *CVPR*, 2019. 8

[6] Zhongang Cai, Mingyuan Zhang, Jiawei Ren, Chen Wei, Daxuan Ren, Jiatong Li, Zhengyu Lin, Haiyu Zhao, Shuai Yi, Lei Yang, et al. Playing for 3d human recovery. *arXiv preprint arXiv:2110.07588*, 2021. 2

[7] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017. 3, 4

[8] Xingyu Chen, Yufeng Liu, Dong Yajiao, Xiong Zhang, Chongyang Ma, Yanmin Xiong, Yuan Zhang, and Xiaoyan Guo. Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image. In *CVPR*, 2022. 8

[9] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018. 4

[10] Bowen Cheng, Bin Xiao, Jingdong Wang, Honghui Shi, Thomas S Huang, and Lei Zhang. Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation. In *CVPR*, 2020. 3, 4, 8

[11] Junhyeong Cho, Kim Youwang, and Tae-Hyun Oh. Cross-attention of disentangled modalities for 3d human mesh recovery with transformers. In *ECCV*, 2022. 1, 2, 4, 5, 8

[12] Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond static features for temporally consistent 3d human pose and shape from a video. In *CVPR*, 2021. 2

[13] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2mesh: Graph convolutional network for 3d human pose and mesh recovery from a 2d human pose. In *ECCV*, 2020. 1, 2, 4, 5, 8

[14] Hai Ci, Chunyu Wang, Xiaoxuan Ma, and Yizhou Wang. Optimizing network structure for 3d human pose estimation. In *ICCV*, 2019. 1

[15] 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. 4

[16] Salehe Erfanian Ebadi, Saurav Dhakad, Sanjay Vishwakarma, Chunpu Wang, You-Cyuan Jhang, Maciek Chociej, Adam Crespi, Alex Thaman, and Sujoy Ganguly. Psp-hdri +: A synthetic dataset generator for pre-training of human-centric computer vision models. *arXiv preprint arXiv:2207.05025*, 2022. 2

[17] Peng Guan, Alexander Weiss, Alexandru O Balan, and Michael J Black. Estimating human shape and pose from a single image. In *ICCV*, 2009. 2

[18] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *CVPR*, 2018. 2

[19] Yana Hasson, Gul Varol, Dimitrios Tzionas, Igor Kalevatykh, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning joint reconstruction of hands and manipulated objects. In *CVPR*, 2019. 8

[20] 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, 2014. 5

[21] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. In defense of grid features for visual question answering. In *CVPR*, 2020. 4

[22] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *3DV*, 2021. 2

[23] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 1, 2, 4, 5, 6

[24] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. In *CVPR*, 2020. 2, 3, 5

[25] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *ICCV*, 2021. 1, 5

[26] 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 *ICCV*, 2019. 1, 2, 4, 5

[27] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019. 1, 2, 4, 5

[28] 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 *CVPR*, 2017. 2, 5

[29] 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 *ECCV*, 2022. 1, 2, 4, 5, 7, 8

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

[31] Kevin Lin, Lijuan Wang, and Zicheng Liu. Mesh graphormer. In *ICCV*, 2021. 1, 2, 4, 5, 6, 8

[32] 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 *ECCV*, 2014. 3, 5

[33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 4

[34] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics*, 34(6):248, 2015. 2, 5

[35] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *ICCV*, 2019. 1, 2, 3, 5

[36] Christian Mandery, Ömer Terlemez, Martin Do, Nikolaus Vahrenkamp, and Tamim Asfour. The kit whole-body human motion database. In *ICAR*, 2015. 2

[37] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2017. 1

[38] 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 *3DV*, 2018. 5

[39] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Neuralannot: Neural annotator for 3d human mesh training sets. In *CVPR Workshop*, 2022. 2

[40] Gyeongsik Moon and Kyoung Mu Lee. I2l-meshnet: Image-to-lixel prediction network for accurate 3d human pose and mesh estimation from a single rgb image. In *ECCV*, 2020. 1, 2, 4, 5, 8

[41] Gyeongsik Moon and Kyoung Mu Lee. Pose2pose: 3d positional pose-guided 3d rotational pose prediction for expressive 3d human pose and mesh estimation. *arXiv preprint arXiv:2011.11534*, 2020. 2

[42] Meinard Müller, Andreas Baak, and Hans-Peter Seidel. Efficient and robust annotation of motion capture data. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2009. 2

[43] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 4

[44] Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Human mesh recovery from multiple shots. *arXiv preprint arXiv:2012.09843*, 2020. 2

[45] Georgios Pavlakos, Luyang Zhu, Xiaowei Zhou, and Kostas Daniilidis. Learning to estimate 3d human pose and shape from a single color image. In *CVPR*, 2018. 1, 2, 5

[46] Mathis Petrovich, Michael J Black, and Gül Varol. Temos: Generating diverse human motions from textual descriptions. *arXiv preprint arXiv:2204.14109*, 2022. 2

[47] Grégory Rogez and Cordelia Schmid. Mocap-guided data augmentation for 3d pose estimation in the wild. *NeurIPS*, 2016. 2, 3

[48] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 3, 4

[49] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Michael J Black, and Tao Mei. Monocular, one-stage, regression of multiple 3d people. In *ICCV*, 2021. 5

[50] Xiao Tang, Tianyu Wang, and Chi-Wing Fu. Towards accurate alignment in real-time 3d hand-mesh reconstruction. In *ICCV*, 2021. 8

[51] Guy Tevet, Brian Gordon, Amir Hertz, Amit H Bermano, and Daniel Cohen-Or. Motionclip: Exposing human motion generation to clip space. *arXiv preprint arXiv:2203.08063*, 2022. 2

[52] Guy Tevet, Sigal Raab, Brian Gordon, Yonatan Shafir, Daniel Cohen-Or, and Amit H Bermano. Human motion diffusion model. *arXiv preprint arXiv:2209.14916*, 2022. 2

[53] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, 2017. 2

[54] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *CVPR Workshop*, 2018. 2

[55] Nikolaus F Troje. Decomposing biological motion: A framework for analysis and synthesis of human gait patterns. *Journal of vision*, 2(5):2–2, 2002. 2

[56] Hsiao-Yu Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *NeurIPS*, 2017. 2

[57] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 2, 3

[58] Timo von Marcard, Roberto Henschel, Michael Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *ECCV*, 2018. 5

[59] Ziniu Wan, Zhengjia Li, Maoqing Tian, Jianbo Liu, Shuai Yi, and Hongsheng Li. Encoder-decoder with multi-level attention for 3d human shape and pose estimation. In *ICCV*, 2021. 2

[60] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 3, 4

[61] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 4, 8

[62] Mihai Zanfir, Andrei Zanfir, Eduard Gabriel Bazavan, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Thundr: Transformer-based 3d human reconstruction with markers. In *ICCV*, 2021. 5

[63] Hongwen Zhang, Jie Cao, Guo Lu, Wanli Ouyang, and Zhenan Sun. Learning 3d human shape and pose from dense body parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020. 2

[64] 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 *ICCV*, 2021. 5

[65] Ce Zheng, Sijie Zhu, Matias Mendieta, Taojiannan Yang, Chen Chen, and Zhengming Ding. 3d human pose estimation with spatial and temporal transformers. In *ICCV*, 2021. 1

[66] Xiaowei Zhou, Menglong Zhu, Georgios Pavlakos, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 5

[67] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *ICCV*, 2019. 8