Pose2Pose: 3D Positional Pose-Guided 3D Rotational Pose Prediction for Expressive 3D Human Pose and Mesh Estimation

Gyeongsik Moon Kyoung Mu Lee

ASRI, Seoul National University, Korea

{mks0601, kyoungmu}@snu.ac.kr

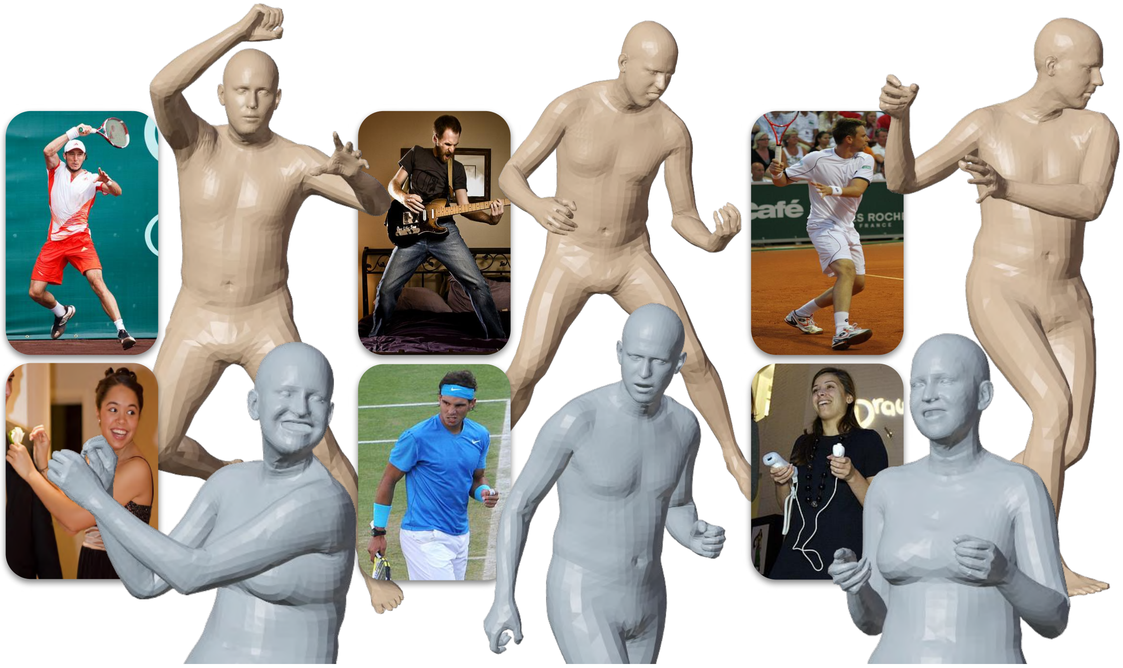

Figure 1: Qualitative results of the proposed Pose2Pose on MSCOCO [26]. Pose2Pose can produce accurate expressive 3D human pose and mesh, which includes body, hands, and face. The gender is only used for the visualization.

Abstract

Previous expressive 3D human pose and mesh estimation methods mostly rely on a single image feature vector to predict 3D rotations of human joints (i.e., 3D rotational pose) from an input image. However, the single image feature vector lacks human joint-level features. To resolve the limitation, we present Pose2Pose, a 3D positional pose-guided 3D rotational pose prediction framework for expressive 3D human pose and mesh estimation. Pose2Pose extracts the joint-level features on the position of human joints (i.e., positional pose) using a positional pose-guided pooling, and the joint-level features are used for the 3D rotational pose prediction. Our Pose2Pose is trained in an end-to-end manner and largely outperforms previous expressive methods. The codes will be publicly available.

1. Introduction

Expressive 3D human pose and mesh estimation aims to localize joints and mesh vertices of all human parts, including body, hands, and face, simultaneously in the 3D space. By combining 3D pose and mesh of all human parts, we can understand not only human articulation and shape but also human intention and feeling, which can be useful in motion capture, virtual/augmented reality, and human action recognition. This is a very challenging task and has been addressed by few recent approaches.

Previous expressive 3D human pose and mesh estimation methods [6] rely on only a single image feature vector to predict 3D rotations of human joints (*i.e.*, 3D rotational pose). They perform global average pooling (GAP) on the extracted image feature from a backbone netowrk [14, 37] and pass the pooled feature to several fully-connected layers for the 3D rotational pose prediction. The estimated 3D rotations are passed to human model layers (*e.g.*, SMPL [27] for body, MANO [33] for hands, FLAME [24] for face, or SMPL-X [31] for all parts) for the final 3D pose and mesh. Although the image feature vector contains an instance-level feature, it lacks joint-level features, which can be obtained from features on the position of human joints (*i.e.*, positional pose). However, GAP in their networks breaks the spatial domain; thus, it limits the chance of utilizing the joint-level features on the positional pose.

To resolve the limitation, we present *Pose2Pose*, a 3D positional pose-guided 3D rotational pose prediction framework for expressive 3D human pose and mesh estimation. Our Pose2Pose consists of PositionNet and RotationNet. PositionNet predicts the 3D positional pose from an input image in a fully-convolutional way. Then, a positional pose-guided pooling extracts the joint-level features on the predicted positional pose of the ResNet output image feature. The RotationNet predicts 3D rotational pose from the 3D positional pose and joint-level features.

Although its effectiveness, it is not trivial to utilize the 3D positional pose-guided 3D rotational pose prediction scheme for the expressive 3D human pose and mesh estimation. Previous expressive methods [6], based on GAP, predict initial 3D hands and face from a body image. From the initial ones, they make hands and face boxes, which crop the hands and face images from a high-resolution body image, respectively. Then, separated networks refine the initial ones by taking the cropped images. We observed that there are two weaknesses in this previous approach. First, predicting initial 3D hands and face from a body image severely hurts the 3D body accuracy. The small sizes of them make it hard to correctly extract useful joint-level features on the positional pose; therefore, gradients calculated from the initial 3D hands and face prediction give a huge burden to the system. Second, their hand refinement network often produces implausible 3D wrist rotations when hands are occluded. This is because their 3D wrist rotation refinement is performed from a cropped hand image, which does not contain global context of the body.

Therefore, we propose to remove the combination of initial 3D hands/face prediction and the refinement. To this end, we design our Pose2Pose as a combination of body, hand, and face branches, where the body branch does not predict initial 3D hands and face. By removing the initial 3D hands and face prediction in the body branch, we successfully train Pose2Pose without hurting the 3D body accuracy. We note that the removal marginally affects final 3D hands and face accuracy as the initial ones mostly carry very rough 3D hands and face. In addition, we predict 3D wrist rotations using the body and hand joint-level features together, extracted by the proposed positional pose-guided pooling from the body and hand-cropped features, respectively. The body joint-level feature allows our Pose2Pose to utilize global context of the body as it is computed from the body image; therefore, our Pose2Pose produces plausible 3D wrist rotations even when the hands are severely occluded. Furthermore, the hand joint-level features significantly boost the 3D wrist rotation accuracy as body joints alone cannot decide the 3D wrist rotations. Our Pose2Pose is trained in an end-to-end manner and significantly outperforms previous expressive 3D human pose and mesh estimation methods. Figure 1 shows qualitative results of the proposed Pose2Pose.

Our contributions can be summarized as follows.

- We present Pose2Pose, a 3D positional pose-guided 3D rotational pose prediction framework for expressive 3D human pose and mesh estimation. Unlike previous approaches that used only an instance-level feature, Pose2Pose uses joint-level features.
- We remove the initial 3D hands/face prediction and the refinement, used in previous expressive method [6], to remove a bad effect on the body branch optimization, which arises from the small image areas of hands and face. In addition, utilizing the body and hand joint-level features together allows Pose2Pose to produce plausible and accurate 3D wrist rotations even when hands are severely occluded.
- Our Pose2Pose is trained in an end-to-end manner and largely outperforms previous expressive 3D human pose and mesh estimation methods.

2. Related works

Part-specific 3D human pose and mesh estimation. Part-specific 3D human pose and mesh estimation methods recover one of 3D body, hands, and face. For the body part, SMPL [27] 3D body model is widely used, parameterized by pose (*i.e.*, 3D rotations of body joints) and shape (*e.g.*, fat/thin and short/tall) parameters. Kanazawa *et al.* [18] proposed an end-to-end trainable human mesh recovery system that uses the adversarial loss to make their output human shape is anatomically plausible. Pavlakos *et al.* [32] used 2D joint heatmaps and silhouette as cues for predicting accurate SMPL parameters. Guler *et al.* [10] used a voting scheme to predict 3D rotations of body joints. Kolotouros *et al.* [21] introduced a self-improving system consists of the SMPL parameter regressor and iterative fitting framework [3]. Moon and Lee [28] proposed an image-to-lixel prediction network, which predicts lixel-based 1D heatmaps for each joint or mesh vertex. Choi *et al.* [5] presented a graph convolutional system that recovers 3D human mesh vertices coordinates from a 2D human pose. Song *et al.* [36] proposed a network that learns to fit their predicted 3D pose and mesh to a target 2D pose.

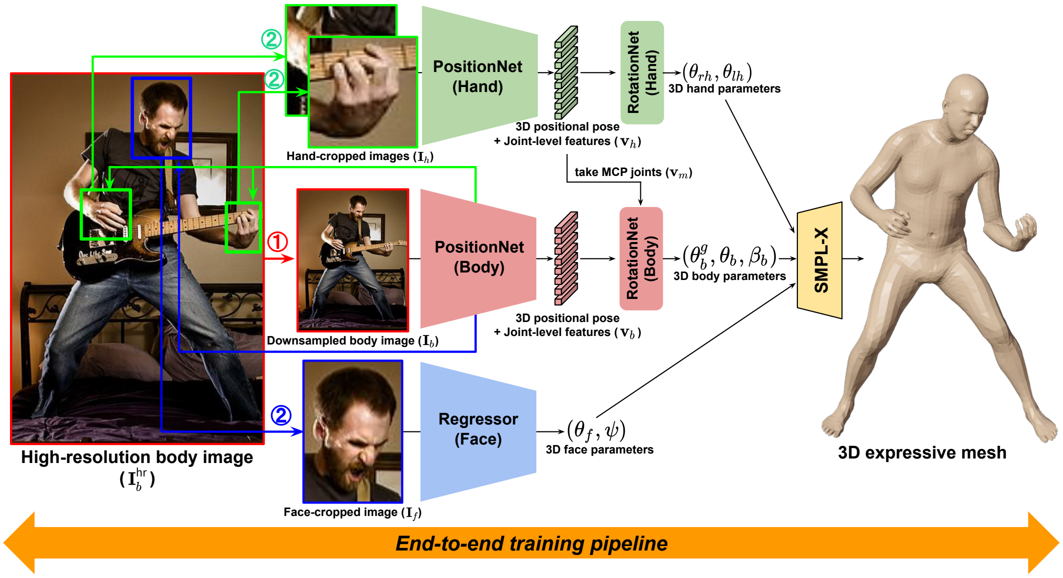
②
②
②
①
PositionNet (Hand)
RotationNet (Hand)
$\to (\theta_{rh}, \theta_{lh})$
3D hand parameters
3D positional pose + Joint-level features ($\mathbf{v}_h$)
Hand-cropped images ($\mathbf{I}_h$)
take MCP joints ($\mathbf{v}_m$)
PositionNet (Body)
RotationNet (Body)
$\to (\theta_b^g, \theta_b, \beta_b)$
3D body parameters
SMPL-X
3D positional pose + Joint-level features ($\mathbf{v}_b$)
Downsampled body image ($\mathbf{I}_b$)
Regressor (Face)
$\to (\theta_f, \psi)$
3D face parameters
High-resolution body image ($\mathbf{I}_b^{\text{hr}}$)
Face-cropped image ($\mathbf{I}_f$)
3D expressive mesh
End-to-end training pipeline

Figure 2: The overall pipeline of Pose2Pose for expressive 3D human pose and mesh estimation. First, the body branch predicts 3D body and boxes of hands and face. Then, the hand and face branches predict 3D hands and face by taking the cropped images. The final expressive 3D human pose and mesh is obtained by forwarding the outputs of body, hand, and face branches to the SMPL-X [31] layer. Our Pose2Pose is trained in an end-to-end manner.

For the hand part, MANO [33] 3D hand model is widely used, parameterized by pose (*i.e.*, 3D rotations of hand joints) and shape (*e.g.*, fat/thin, small/big) parameters. Baek *et al.* [2] trained their network to estimate the MANO parameters using a differentiable renderer. Boukhayma *et al.* [4] trained their network that takes a single RGB image and estimates MANO parameters by minimizing the distance of the estimated hand joint locations and groundtruth. Ge *et al.* [9] proposed a GraphCNN, which directly estimates vertices coordinates of hand mesh from the extracted image feature. Kulon *et al.* [23] presented a weakly-supervised mesh-convolutional hand reconstruction system, which leverages in-the-wild video dataset. Moon and Lee [28] and Choi *et al.* [5] showed their networks also perform well not only on the human body but also on the human hand.

For the face part, FLAME [24] 3D face model is widely used, parameterized by pose (*i.e.*, 3D rotations of jaw and eyes) and facial expression parameters. Sela *et al.* [35] uses a synthetic dataset to make an image-to-depth mapping and a pixel-to-vertex mapping. The mappings are combined to generate the face mesh. Tuan *et al.* [40] fit a 3D morphable model to multiple images of the same subject to generate groundtruth 3D face mesh. Tewari *et al.* [39] train their network in an end-to-end manner using a photometric loss and an optional 2D feature loss. Sanyal *et al.* [34] proposed RingNet, which explicitly enforces the identity consistency between the estimated identity codes from the multiple images of the same subject.

Expressive 3D human pose and mesh estimation. Due to its difficulty and absence of the unified expressive body model, there have been very few attempts to simultaneously recover the 3D human pose and mesh of all human parts, including body, hands, and face. Most previous attempts are an optimization-based approach, which fits a 3D human model to the 2D/3D evidence. Joo *et al.* [17] fits their human models (*i.e.*, Frank and Adam) to 3D human joints coordinates and point clouds in a multi-view studio environment. Xiang *et al.* [42] extended Joo *et al.* [17] to the single RGB case. Pavlakos *et al.* [31] and Xu *et al.* [44] fits their human model, SMPL-X and GHUM, respectively, to 2D human joint coordinates. As the above optimization-based methods can be slow and prone to noisy evidence, a regression-based approach is presented recently. Choutas *et al.* [6] presented ExPose, which consists of body, hand, and face networks. Their body network predicts initial 3D hands and face, and boxes of hands and face are made by projecting the initial ones to the 2D space. Then, the separated hand and face networks refine the initial ones.

Our Pose2Pose is also the regression-based approach; however, it has clear three differences compared with the previous work, ExPose [6]. First, Pose2Pose utilizes joint-level features by the positional pose-guided pooling, while ExPose utilizes only instance-level features by the GAP. Second, we remove the initial 3D hands and face prediction from a body image, which has a bad effect on the 3D body accuracy due to their small areas in a body image. Finally, we predict 3D wrist rotations from body and hand joint-level features together, which allows Pose2Pose to produce plausible and accurate rotations even when the hands are severely occluded.

Utilizing joint-level features for 3D human pose and mesh estimation. Guler *et al.* [10] proposed a system that predicts 3D joint rotation cluster weights from joint-level features. By aggregating the predicted weights, they obtain the final 3D joint rotations. This voting scheme can prevent implausible 3D joint rotations as the 3D joint rotation clusters are pre-defined from datasets. Zhang *et al.* [45] utilizes UVI maps of human body parts and graph convolutional network to predict 3D joint rotations.

Our Pose2Pose has three distinctive points compared with the above methods. First, Pose2Pose uses both joint-level image features and 3D geometric evidence, which brings noticeable performance gain. On the other hand, Guler *et al.* [10] use only joint-level image features, and Zhang *et al.* [45] use only 3D geometric evidence (*i.e.*, body part UVI maps). Second, they require DensePose dataset [11], which contains dense correspondence between pixels and 3D mesh vertices, to train their networks, while our Pose2Pose does not require it. In particular, DensePose dataset is greatly challenging to collect for the expressive 3D human pose and mesh estimation due to the small hands and face image areas. Finally, Pose2Pose can recover expressive 3D pose and mesh, while the above methods are designed only for the 3D body recovery.

3. Pose2Pose

Figure 2 shows the overall pipeline of the proposed Pose2Pose for expressive 3D human pose and mesh estimation. It consists of body, hand, and face branches, which take a cropped body, hands, and face images, respectively. The outputs of each branch (*i.e.*, 3D human model parameters of each part) are fed to SMPL-X [31] layer to obtain the final expressive 3D human pose and mesh. We provide a detailed description of each branch below.

3.1. Body branch

The body branch consists of PositionNet and RotationNet, and the two networks are connected by the positional pose-guided pooling. Unlike ExPose [6], the body branch of our Pose2Pose does not predict initial 3D hands and face as the small size of them in the body image, shown in Figure 3, has a bad effect on optimizing the body branch.

PositionNet. The PositionNet of the body branch takes a body image $\mathbf{I}_b$, downsampled from a high-resolution body image $\mathbf{I}_b^{\text{hr}}$, and predicts 3D positional pose (*i.e.*, 3D positions of human joints) of human body $\mathbf{P}_b = [\mathbf{p}_{b,1}, \dots, \mathbf{p}_{b,J_b}]^T \in \mathbb{R}^{J_b \times 3}$. J_b denotes the number of body joints. x- and y-axis of $\mathbf{P}_b$ are in image space, and z-axis of it is in root joint (*i.e.*, pelvis)-relative depth space. To this end, the PositionNet extracts image feature $\mathbf{F}_b \in \mathbb{R}^{2048 \times H_b \times W_b}$ from the input image using ResNet-50 [14], where H_b and W_b denote the height and width of $\mathbf{F}_b$, respectively. We use ResNet after removing the GAP and fully-connected layer of the last part of the original ResNet. Then, a 1-by-1 convolution predicts 3D heatmaps of human body joints $\mathbf{H}_b \in \mathbb{R}^{J_b \times D \times H_b \times W_b}$. To predict the 3D heatmaps from the 2D feature map $\mathbf{F}_b$, the 1-by-1 convolution first predicts a tensor of shape $\mathbb{R}^{J_b D \times H_b \times W_b}$, and we reshape the tensor to the shape of $\mathbf{H}_b$ following Sun *et al.* [38]. The 3D positional pose $\mathbf{P}_b$ is calculated from $\mathbf{H}_b$ by the soft-argmax operation [38] in a differentiable way. The shape parameter $\beta_b \in \mathbb{R}^{10}$ and 3D global translation vector $\mathbf{t}_b \in \mathbb{R}^3$ are predicted from the image feature $\mathbf{F}_b$ using the GAP and a single fully-connected layer.

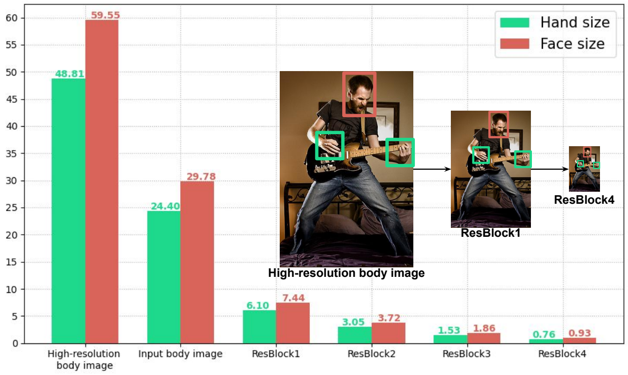

Figure 3: The average hands and face sizes of MSCOCO [16] training set in the high-resolution body image $\mathbf{I}_b^{\text{hr}}$, input body image $\mathbf{I}_b$, and outputs of each block of ResNet, where each has 512 × 384, 256 × 192, 64 × 48, 32 × 24, 16 × 12, and 8 × 6 resolution, respectively. The size represents the average width and height of the bounding box.

Positional pose-guided pooling. The positional pose-guided pooling computes the joint-level features $\mathbf{F}_b^P = [\mathbf{f}_{b,1}^P, \dots, \mathbf{f}_{b,J_b}^P]^T \in \mathbb{R}^{J_b \times 512}$ using the predicted 3D positional pose $\mathbf{P}_b$, as illustrated in Figure 4. To this end, we obtain the jth joint feature $\mathbf{f}_{b,j}^P$ at (x, y) position of $\mathbf{p}_{b,j}$ using bilinear interpolation on the image feature map $\mathbf{F}'_b$. $\mathbf{F}'_b$ is obtained by applying a 1-by-1 convolutional layer, which changes the channel dimension from 2048 to 512, to $\mathbf{F}_b$. The interpolated feature $\mathbf{f}_{b,j}^P$ provides global contextual information around the jth joint thanks to the large receptive field size of ResNet.

RotationNet. The RotationNet of the body branch takes a vector $\mathbf{v}_b$, a concatenation of the flattened 3D positional pose $\mathbf{P}_b$ and flattened joint-level features $\mathbf{F}_b^P$. The 3D positional pose provides 3D geometric evidence, while the joint-level features provide global contextual information around human joints. As the body joints do not include the hand joints, additional hand joints are necessary for accurate 3D elbow and wrist rotation recovery. Therefore, the RotationNet of the body branch additionally takes a vector $\mathbf{v}_m$, a concatenation of the flattened 3D positional pose and flat-

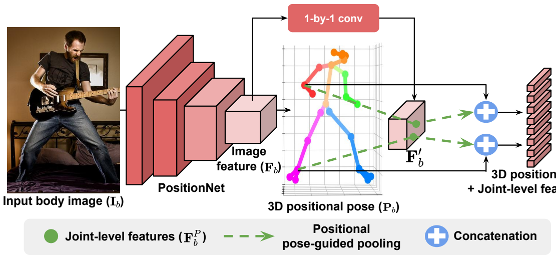
1-by-1 conv
Input body image ($\mathbf{I}_b$)
PositionNet
Image feature ($\mathbf{F}_b$)
3D positional pose ($\mathbf{P}_b$)
$\mathbf{F}'_b$
3D positional pose + Joint-level features ($\mathbf{v}_b$)
● Joint-level features ($\mathbf{F}_b^P$)
Positional pose-guided pooling
Concatenation

Figure 4: Illustration of the positional pose-guided pooling in the body branch. It extracts the joint-level features by performing the bilinear interpolation at (x, y) position of the 3D positional pose $\mathbf{P}_b$ on the image feature $\mathbf{F}'_b$. For the simplicity, we describe only right elbow and ankle.

tened joint-level features of the hand MCP joints, obtained from the hand branch. We provide how $\mathbf{v}_m$ is obtained from the hand branch in Section 3.2. From the concatenation of $\mathbf{v}_b$ and $\mathbf{v}_m$, the RotationNet predicts the 3D rotational pose of body (*i.e.*, 3D rotations of human body joints), which includes 3D body global rotation $\theta_b^g \in \mathbb{R}^3$ and 3D body joint rotations $\theta_b \in \mathbb{R}^{21\times3}$ using a single fully-connected layer. θ_b includes 3D rotations of wrists, neck, and head.

Hands and face box prediction. The body branch predicts hand and face bounding boxes by concatenating the image feature $\mathbf{F}_b$ and 2D heatmap $\mathbf{H}'_b$ and passing it to two convolutional layers. The 2D heatmap $\mathbf{H}'_b$ is generated by making a Gaussian blob on the (x, y) position of $\mathbf{P}_b$. The soft-argmax [38] is applied to the output of the convolutional layers for the box centers. The widths and heights of the boxes are computed by performing positional pose-guided pooling on the box centers of $\mathbf{F}_b$ and pass the features of each box center to separated fully-connected layers.

3.2. Hand branch

Like the body branch, the hand branch consists of the PositionNet and RotationNet, and the two networks are connected by the positional pose-guided pooling.

PositionNet. Likewise the body branch, the hand branch takes a hand-cropped image $\mathbf{I}_h$ and predicts 3D positional pose of hand joints $\mathbf{P}_h = [\mathbf{p}_{h,1}, \dots, \mathbf{p}_{h,J_h}]^T \in \mathbb{R}^{J_h\times3}$, where J_h denotes the number of single hand joints. x- and y-axis of $\mathbf{P}_h$ are in image space, and z-axis of it is in root joint (*i.e.*, wrist)-relative depth space. To this end, the hand image $\mathbf{I}_h$ is cropped and resized from the high-resolution body image $\mathbf{I}_b^{\text{hr}}$ by applying RoIAlign [13] at the predicted hand bounding box area. Taking the hand images from the high-resolution image allows Pose2Pose to utilize detailed information, essential for the highly articulated 3D hands prediction. The hand-cropped images of the left and right hands are flipped to the right hands and are fed to the PositionNet. The PositionNet extracts the image feature $\mathbf{F}_h \in \mathbb{R}^{2048\times H_h\times W_h}$ from the hand-cropped images $\mathbf{I}_h$ using ResNet-50 [14], where H_h and W_h denote the height and width of the $\mathbf{F}_h$, respectively. We use ResNet after removing the GAP and fully-connected layer of the last part of the original ResNet. Like the body branch, the combination of a 1-by-1 convolutional layer and soft-argmax is used to obtain the 3D positional pose of hand joints.

Positional pose-guided pooling. Likewise the body branch, the positional pose-guided pooling extracts the hand joint-level features $\mathbf{F}_h^P = [\mathbf{f}_{h,1}^P, \dots, \mathbf{f}_{h,J_h}^P]^T \in \mathbb{R}^{J_h\times512}$ at (x, y) position of $\mathbf{P}_h$ on $\mathbf{F}_h$.

RotationNet. The RotationNet of the hand branch takes a vector $\mathbf{v}_h$, a concatenation of the flattened 3D positional pose of hand $\mathbf{P}_h$ and flattened hand joint-level features $\mathbf{F}_h^P$. From the vector $\mathbf{v}_h$, the RotationNet of the hand branch predicts the 3D rotational pose of hand (*i.e.*, 3D hand joint rotations) $\theta_h \in \mathbb{R}^{15\times3}$ using a fully-connected layer. θ_h includes 3D finger rotations without a 3D wrist rotation. After the prediction, we flip back the flipped left hands; therefore, $\mathbf{P}_h$ is split to $\mathbf{P}_{lh}$ and $\mathbf{P}_{rh}$, and θ_h is split to θ_{lh} and θ_{rh}. lh and rh denote left and right hands, respectively.

MCP joints to the body branch. As described in Section 3.1, we pass the MCP joint features to the body branch for the accurate 3D elbow and wrist rotation prediction. To this end, we take the 3D positional pose of MCP joints from that of all hand joints $\mathbf{P}_h$. Likewise, we take the joint-level features of MCP joints from that of all hand joints $\mathbf{F}_h^P$. The taken 3D positional pose and joint-level features are concatenated and flattened to a vector $\mathbf{v}_m$, which is passed to the body branch. Among hand joints, we choose the MCP joints as they are easier to predict than other hand joints while providing 3D elbow and wrist rotation information.

3.3. Face branch

Unlike the body and hands, face keypoints do not move according to 3D rotations of joints. Therefore, instead of utilizing the 3D positional pose-guided 3D rotational pose prediction scheme, we design the face branch as a simple combination of ResNet-18 and a fully-connected layer, which takes a face-cropped image $\mathbf{I}_f$ and predicts 3D jaw rotation $\theta_f \in \mathbb{R}^3$ and facial expression code $\psi \in \mathbb{R}^{10}$. Like the hand branch, the face-cropped image $\mathbf{I}_f$ is cropped and resized from the high-resolution body image $\mathbf{I}_b^{\text{hr}}$ by applying RoIAlign [13] at the predicted face bounding box area. Taking the face images from the high-resolution image preserves detailed facial expressions. We use ResNet after removing the GAP and fully-connected layer of the last part of the original ResNet.

3.4. Loss functions

Our framework is trained in an end-to-end manner by minimizing the loss function L, defined as follows.

$$L = L_{\text{param}} + L_{\text{coord}} + L_{\text{box}}, \tag{1}$$

where L_{param} is a $L1$ distance between predicted and groundtruth SMPL-X parameters. L_{coord} is a $L1$ distance between predicted and groundtruth joint coordinates, and three types of joint coordinates are used to calculate the loss function: 1) 3D positional pose of the body $\mathbf{P}_b$ and hands $\mathbf{P}_h$, 2) 3D coordinates from the 3D mesh, and 3) 2D coordinates, obtained by projecting the 3D coordinates from the 3D mesh, to the 2D space using the perspective projection. For the projection, the predicted 3D global translation vector $\mathbf{t}_b$, fixed focal length (1500,1500), and fixed principal points (*i.e.*, a center point of $\mathbf{I}_b$) are used, following [21]. We observed that the perspective projection provides slightly better accuracy than the orthogonal projection [18]. Finally, L_{box} is a $L1$ distance between predicted and groundtruth center and scale of hands and face boxes.

4. Implementation details

PyTorch [30] is used for implementation. The ResNet of the body branch is initialized with that of Xiao *et al.* [43], pre-trained on MSCOCO 2D human pose dataset. The hand branch is pre-trained on FreiHAND [48] and the whole-body version of MSCOCO [16]. The remaining parts are randomly initialized. The weights are updated by Adam optimizer [20] with a mini-batch size of 96. The size of the high-resolution body image $\mathbf{I}_b^{\text{hr}}$, downsampled body image $\mathbf{I}_b$, hand image $\mathbf{I}_h$, and face image $\mathbf{I}_f$ are 512×384, 256×192, 256×256, and 192×128, respectively. Data augmentations, including scaling, rotation, random horizontal flip, and color jittering, are performed in training. All the 3D rotations are initially predicted in the 6D rotational representation of Zhou *et al.* [47] and converted to the 3D axis-angle rotations. The initial learning rate is set to 10^{-4} and reduced by a factor of 10 at the 10th epoch. The hand branch is pre-trained for 12 epochs with four NVIDIA RTX 2080 Ti GPUs, which take 4 hours. Then, the whole framework is trained in an end-to-end manner for 12 epochs with four NVIDIA RTX 2080 Ti GPUs, which take 1 day.

5. Experiment

5.1. Datasets and evaluation metrics

Datasets. For the expressive 3D human pose and mesh estimation, Human3.6M [15], whole-body version of MSCOCO [16], and MPII [1] are used for the training, and EHF [31] is used for the testing. We provide qualitative results on MSCOCO validation set. NeuralAnnot [29] is used to obtain 3D pseudo-GT SMPL-X fits of the training sets.

Evaluation metrics. MPJPE and MPVPE are used to evaluate 3D pose and mesh, respectively, where each calculates the average 3D joint distance (mm) and 3D mesh vertex distance (mm) between predicted and groundtruth, after aligning a root joint translation. PA MPJPE and PA MPVPE further align a rotation and scale.

Settings	All	Body	Hands	Face	Mem.	Time
GAP [6]	54.8	66.4	12.7	**5.8**	**10.2 GB**	**0.22 sec.**
GAP+PPP	52.2	62.3	12.8	**5.8**	10.4 GB	0.24 sec.
PPP (Ours)	**50.7**	**60.9**	**11.4**	**5.8**	10.3 GB	0.23 sec.

Table 1: 3D error, GPU memory usage, and forward time of each iteration comparison between models that use GAP and PPP on EHF.

Settings	All	Body	Hands	Face
Body-only Pose2Pose	-	**60.6**	-	-
With initial prediction	59.8	70.5	12.7	6.1
With initial prediction + refine [6]	58.5	68.9	12.6	6.0
Without initial prediction (Ours)	**50.7**	60.9	**11.4**	**5.8**

Table 2: 3D error comparison between models with various expressive 3D human pose and mesh estimation pipeline on EHF.

5.2. Ablation study

For the ablation study, we report 3D errors, including 1) PA MPVPE of all vertices and face part vertices of the 3D mesh and 2) PA MPJPE of body joints and hand joints. The numbers in hands are averaged values of left and right hands.

Benefit of the positional pose-guided pooling. Table 1 shows that the positional pose-guided pooling (PPP) achieves lower 3D errors than GAP, widely used in previous works [6] while consuming almost the same amount of computational cost. This is because the joint-level features, obtained by PPP, contain more beneficial human articulation information than the instance-level feature, obtained by GAP. Interestingly, the combination of GAP and PPP achieves worse results than solely using PPP because of the many unnecessary information, such as backgrounds, in features from the GAP.

The variant with GAP does not predict the positional pose and directly regresses 3D human model parameters from the global average pooled ResNet output using a fully-connected layer. The MCP joint features cannot be obtained from the hand part as joint-level features are not available. Instead, we provide a global average pooled hand feature to the body RotationNet. The RotationNet of the GAP+PPP additionally takes the global average pooled ResNet output feature compared with the RotationNet of the PPP.

Benefit of removing the initial 3D hands and face prediction. Table 2 shows that removing the initial 3D hands and face prediction in the body branch allows our Pose2Pose to preserve the 3D body accuracy of the body-only Pose2Pose, while using it hurts the 3D body accuracy. In addition, the initial 3D hands and face prediction makes hands and face box localization worse, which results in higher 3D hands and face errors. The 3D hands and face errors are calculated from the final 3D hands and face, obtained from the

Inputs of the RotationNet	All	Body	Hands
2D pose	55.8	67.0	12.6
3D pose	54.3	64.3	12.4
Joint-level feat.	52.2	62.7	11.6
2D pose + Joint-level feat.	52.3	62.4	11.6
3D pose + joint-level feat. (Ours)	**50.7**	**60.9**	**11.4**

Table 3: 3D error comparison between models with various input combinations of the RotationNet on EHF.

Settings	All	Body	Hands
Without MCP features	52.7	**60.9**	12.3
With MCP features (Ours)	**50.7**	**60.9**	**11.4**

Table 4: 3D error comparison between models without and with MCP features in their body branch on EHF.

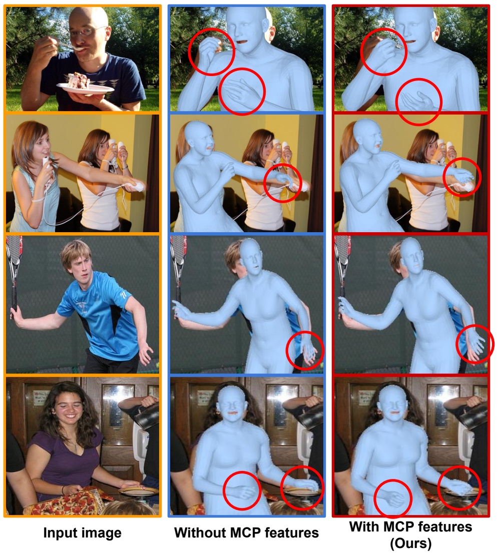

Figure 5: Taking MCP features in the body branch improves the 3D elbow and wrist rotation predictions.

hand and face branches, respectively, not from the initial ones.

For the experiment, we designed all settings to use PPP. The body-only Pose2Pose is trained by only using 3D body groundtruths and setting losses from hands and face to zero. When we predict the initial 3D hands and face, an FPN [25]-style upsampler, a widely used technique for the small object detection, is added after the body branch ResNet to enlarge the small hands and face areas. Both 3D positional and rotational poses in the body branch are predicted from the upsampled feature. The initial 3D hands and face refinement, used in ExPose [6], is performed following their refinement strategy.

Inputs of the RotationNet. Table 3 shows that RotationNet's taking both the 3D positional pose and joint-level image features achieves the lowest 3D errors. The 3D positional pose predicted by PositionNet provides 3D geometric evidence, while the joint-level features extracted by PPP provide contextual information. We design our RotationNet to take both inputs, thus can utilize both geometric evidence and contextual information. In particular, changing PositionNet to predict 2D positional pose and RotationNet to take the 2D one achieves worse performance than our 3D positional pose-based system. This indicates additional depth information of the 3D positional pose plays an important role in the accurate 3D rotational pose prediction. We checked that changing the inputs of the RotationNet does not affect the 3D face error, as the face branch does not contain RotationNet.

Methods	PA MPVPE			PA MPJPE	
	All	Hands	Face	Body	Hands
SMPLify-X [31]	65.3	12.3	6.3	87.6	12.9
MTC [42]	67.2	-	-	107.8	16.7
ExPose [6]	54.5	12.8	**5.8**	62.8	13.1
Pose2Pose (Ours)	**50.3**	**10.8**	**5.8**	**60.4**	**10.8**

Table 5: 3D errors comparison on EHF. The numbers in hands are averaged values of left and right hands.

Benefit of taking MCP features in the body branch. Table 4 and Figure 5 show that taking MCP features in the body branch is necessary for the accurate 3D elbow and wrist rotation prediction. The improved 3D elbow and wrist rotations result in lower 3D mesh error of all vertices in the table. The MCP features also decrease the 3D hand joint error of the table as better 3D elbow and wrist rotation prediction during the training stage makes the global rotation of 3D hands closer to the groundtruth; therefore, our $L1$ loss, calculated between groundtruth and output 2D/3D hand joint coordinates, does not suffer from global rotation misalignment. The MCP features do not change the 3D body joint error of the table as the 3D body joints do not contain roll-axis of the 3D wrist rotation. We checked that changing the MCP features in the body branch does not affect the 3D face error.

5.3. Comparison with state-of-the-art methods

Table 5 shows that our Pose2Pose largely outperforms all previous methods on EHF. Figure 6 shows that ExPose [6] suffers from implausible 3D wrist rotations when hands are occluded because their hand refinement network refines the 3D wrist rotation without global context of the body. For example, the hand-cropped images of the left-top image may contain a piece of the table and cat, which do not provide global context of the body. On the other hand, ours produces highly stable results due to the global context from the body joint-level features. Figure 7 shows that the proposed Pose2Pose achieves better 3D body and hand results. All the qualitative results of ExPose are obtained from their officially released codes. Figure 8 shows several failure cases of Pose2Pose, mainly arise from the depth ambigu-

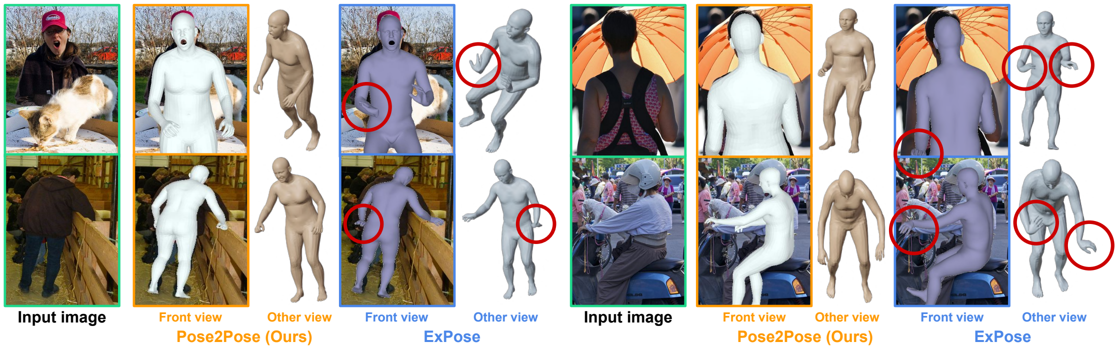

Figure 6: Qualitative results comparisons with ExPose [6] on MSCOCO. Wrong 3D wrist rotations are highlighted.

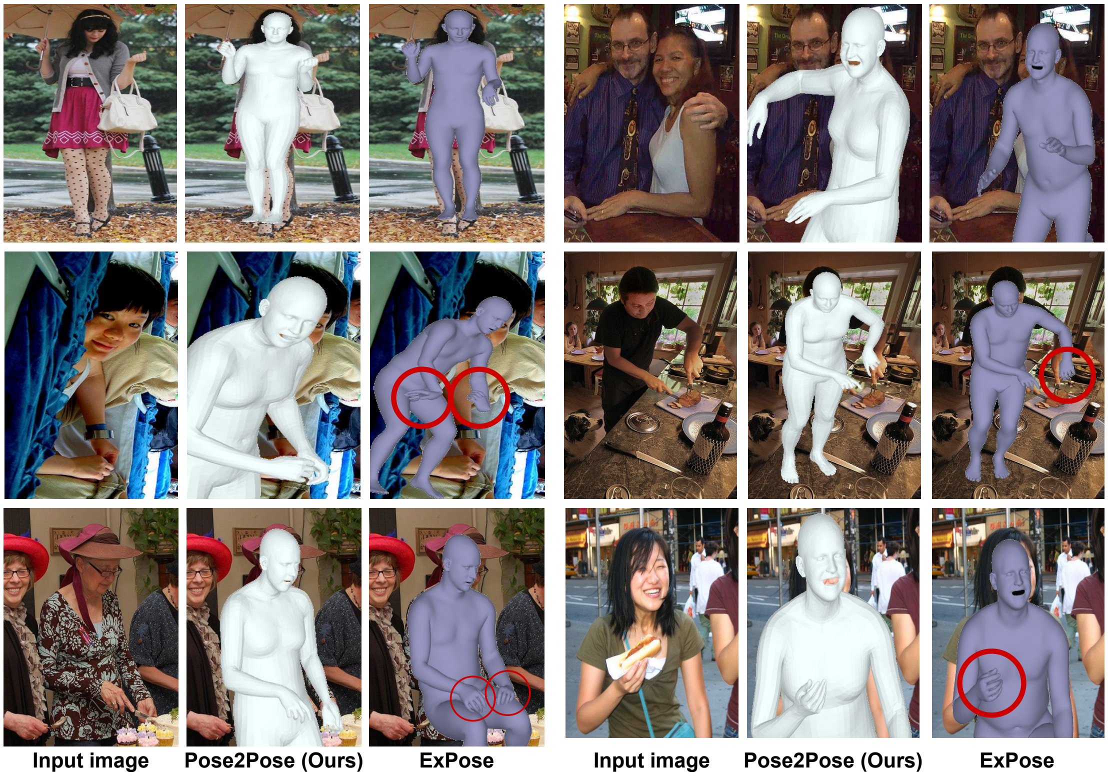

Figure 7: Qualitative results comparisons with ExPose [6] on MSCOCO. Wrong 3D hand results are highlighted.

ity. In particular, recent 3D analysis on human-object interaction [46] can be helpful to resolve the second failure case. Excluding the human detection, both Pose2Pose and ExPose [6] runs at 10 frames per second, measured by using a RTX 2080 Ti GPU and setting the mini-batch size to 1.

6. Conclusion

We present Pose2Pose, a 3D positional pose-guided 3D rotational pose prediction framework for expressive 3D human pose and mesh estimation from a single RGB image. Our Pose2Pose utilizes joint-level features for the accurate 3D rotational pose prediction. In addition, removing the initial 3D hands and face prediction in the body branch allows Pose2Pose to preserve the 3D body accuracy. Finally, ours produces plausible and accurate 3D wrist rotation by utilizing both body and hand joint-level features even when hands are severely occluded. Our Pose2Pose largely outperforms previous methods.

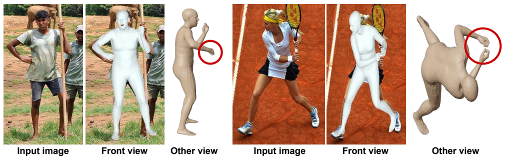

Figure 8: Failure cases of Pose2Pose. Wrong 3D results are highlighted.

Supplementary Material of "Pose2Pose: 3D Positional Pose-Guided 3D Rotational Pose Prediction for Expressive 3D Human Pose and Mesh Estimation"

In this supplementary material, we present more experimental results that could not be included in the main manuscript due to the lack of space.

7. Why Pose2Pose produces plausible 3D hands when hands are occluded?

Figure 6 of the main manuscript shows Pose2Pose produces plausible 3D hands when hands are severely occluded, while ExPose [6] produces implausible ones. We provide two reasons for this below.

3D wrist rotations from both body and MCP features. Pose2Pose produces 3D wrist rotations from both body and MCP features. When hands are occluded in the input image, the MCP features might not contain useful information; however, the body features provide overall body articulation. Therefore, Pose2Pose outputs plausible 3D wrist rotations by seeing the body features when hands are occluded. On the other hand, the hand network of ExPose [6] only takes hand-cropped images and initial 3D hands predictions without body information; therefore, it might produce anatomically implausible 3D wrist rotations when hands are invisible in the hand-cropped images.

3D wrist and finger local rotations. When hands are barely visible in the hand-cropped image due to the occlusions, the hand network of ExPose [6] might predict 3D wrist rotations close to the average of 3D wrist rotations in the training set. The hand network of ExPose [6] outputs 3D wrist global rotations from cropped hand images, where the 3D global rotations represent 3D rotations with respect to a fixed 3D coordinate system (*i.e.*, 3D world coordinate system). The learned ranges of 3D global rotations are very wide as the 3D global rotations are calculated from all the parent joints in the kinematic chain using forward kinematics. Therefore, their hand network produces very random 3D wrist global rotation when hands are invisible in the input hand-cropped image.

On the other hand, Pose2Pose predicts 3D wrist local rotations, which are relative 3D rotations with respect to the parent joints (*i.e.*, elbows). Learning to predict 3D local rotations can effectively limit the range of the output space as the 3D local rotations only consider relative rotations with respect to the parent joints. The limited output space allows our Pose2Pose to produce stable results when hands are invisible in the input hand-cropped images. Please note that simply changing 3D wrist global rotation prediction to 3D wrist local rotation prediction in the hand network of ExPose [6] cannot resolve the limitation perfectly. This is because the hand information should be considered when predicting 3D elbow rotations as rotations in the roll-axis of wrists and elbows are highly correlated. Our Pose2Pose produces 3D body joints rotations from both body and MCP features, and the MCP feature greatly helps to recover not only 3D wrist rotations but also 3D elbow rotations.

Settings	All	Body	Hands	Face	Mem.	Time
Wo. pooling	55.9	63.8	13.0	**5.8**	**10.3 GB**	**0.23 sec.**
PPP (Ours)	**50.7**	**60.9**	**11.4**	**5.8**	**10.3 GB**	**0.23 sec.**

Table 6: 3D error, GPU memory usage, and forward time of each iteration comparison between models without pooling and with PPP on EHF.

Methods	Human3.6M		3DPW	
	MPJPE	PA MPJPE	MPJPE	PA MPJPE
Body-only methods				
HMR [18]	88.8	56.8	130.0	81.3
GraphCMR [22]	-	50.1	-	70.2
SPIN [21]	-	**41.1**	96.9	59.2
Pose2Mesh [5]	64.9	47.0	88.9	58.3
I2L-MeshNet [28]	**55.7**	**41.1**	93.2	57.7
Song *et al*. [36]	-	56.4	-	55.9
Expressive methods				
ExPose [6]	-	-	93.4	60.7
Pose2Pose (Ours)	71.0	47.4	**87.6**	**55.3**

Table 7: 3D body error comparison on Human3.6M and 3DPW.

Both Pose2Pose and ExPose [6] predict 3D finger local rotations, which makes their 3D fingers are anatomically plausible, although hands are invisible in the hand-cropped image.

8. Benefit of PPP

Table 6 shows that a model with the proposed positional pose-guided pooling (PPP) achieves lower 3D errors than a model without pooling. The 3D errors represent 1) PA MPVPE of all vertices and face part vertices of the 3D mesh and 2) PA MPJPE of body joints and hand joints, like those of Table 1 of the main manuscript. Unlike the global average pooling (GAP) removes the spatial domain by averaging, a model without pooling preserves the spatial domain; however, there is much unnecessary information, such as backgrounds, which degrades the performance. On the other hand, our PPP extracts highly useful joint-level features, essential information for the human articulation understanding, which is the reason for lower 3D errors than a model without pooling.

Methods	PA errors	F scores
Hand-only methods		
Hasson *et al.* [12]	13.2 / -	0.436 / 0.908
Boukhayma *et al.* [4]	13.0 / -	0.435 / 0.898
FreiHAND [48]	10.7 / -	0.529 / 0.935
Kulon *et al.* [23]	8.6 / 8.4	0.614 / 0.966
Pose2Mesh [5]	7.8 / 7.7	0.674 / 0.969
I2L-MeshNet [28]	**7.6 / 7.4**	**0.681 / 0.973**
Expressive methods		
ExPose [6]	11.8 / 12.2	0.484 / 0.918
Pose2Pose (ResNet-18)	8.6 / 8.6	0.621 / 0.962
Pose2Pose (Ours)	7.8 / 7.8	0.661 / 0.970

Table 8: 3D hand errors (PA MPVPE/PA MPJPE and F-score@5mm/15mm) comparison on FreiHAND.

Methods	Mean	Median	Std.
Face-only methods			
RingNet [34]	2.08/2.02	1.63/1.58	**1.79/1.68**
Expressive methods			
ExPose [6]	2.27/2.42	1.76/1.91	1.97/2.03
Pose2Pose (Ours)	**2.04/1.98**	**1.57/1.52**	**1.79**/1.76

Table 9: 3D face errors comparison on low-quality/high-quality images of Stirling.

9. Evaluation on part-specific datasets

9.1. Body-only evaluation

Table 7 shows that our body-only Pose2Pose outperforms all body-only and expressive methods on in-the-wild benchmark, 3DPW [41], and achieves comparable results with state-of-the-art methods on in-the-lab benchmark, Human3.6M [15]. For this body evaluation, we trained only body branch of Pose2Pose on Human3.6M [15], MSCOCO [26], and MPII [1]. Following previous works [5, 21, 28], we use SMPL for the human model, and 14 joints are used for the evaluation. Groundtruth boxes are used during the training and testing, following previous works [5, 21, 28].

9.2. Hand-only evaluation

Table 8 shows that our hand-only Pose2Pose achieves comparable accuracy with a recent state-of-the-art hand-only method [28] and significantly outperforms the expressive method [6] on FreiHAND [48]. For a fair comparison with ExPose [6], we additionally report our results using the same backbone with theirs (*i.e.*, ResNet-18 [14]). For this hand part evaluation, we trained only the hand branch of Pose2Pose using MANO 3D hand model on FreiHAND [48] and MSCOCO [16]. Groundtruth boxes are used during the training, and detected boxes by Mask R-CNN [13] are used for the testing.

Where hands/face are cropped from	Hand	Face
Downsampled body image	11.4	6.0
High-resolution body image (Ours)	**10.8**	**5.8**

Table 10: PA MPVPE of hands and face comparison between models that crop hands and face from various sources.

Metric	Hand	Face
IoU	0.54	0.77

Table 11: IoU of the hand and face box on MSCOCO validataion set. The number of the hand is an averaged number of the right and left hands.

9.3. Face-only evaluation

Table 9 shows that our face-only Pose2Pose achieves the lowest errors compared with the face-only method and expressive method [6] on Stirling [8]. For the face part evaluation, we trained only the face branch of Pose2Pose using FLAME 3D face model on FFHQ [19] and MSCOCO [16]. Groundtruth boxes are used during the training, and detected boxes by RetinaFace [7] is used for the testing. The input image size is changed to 256 × 256, following ExPose [6].

10. Benefit of high-resolution body image

Table 10 shows cropping hands and face from the high-resolution body image is necessary for low 3D hands and face errors.

11. Hand and face box localization evaluation

Table 11 shows the intersection over union (IoU) of hand and face bounding boxes, predicted in the body branch of Pose2Pose. As the table shows, ours is good at localizing the face; however, the IoU of the hand part is much lower than that of the face part. Most of the human pose estimation methods have difficulty in accurately localizing hands because of occlusions, small size, and large movement, which should be addressed in future work.

12. Qualitative comparison

Figure 9 and 10 show our Pose2Pose produces more accurate expressive 3D human mesh than ExPose [6]. In particular, ours achieves much better hands results. Figure 11 shows Pose2Pose produces plausible 3D hands when hands are occluded, while ExPose [6] produces implausible ones.

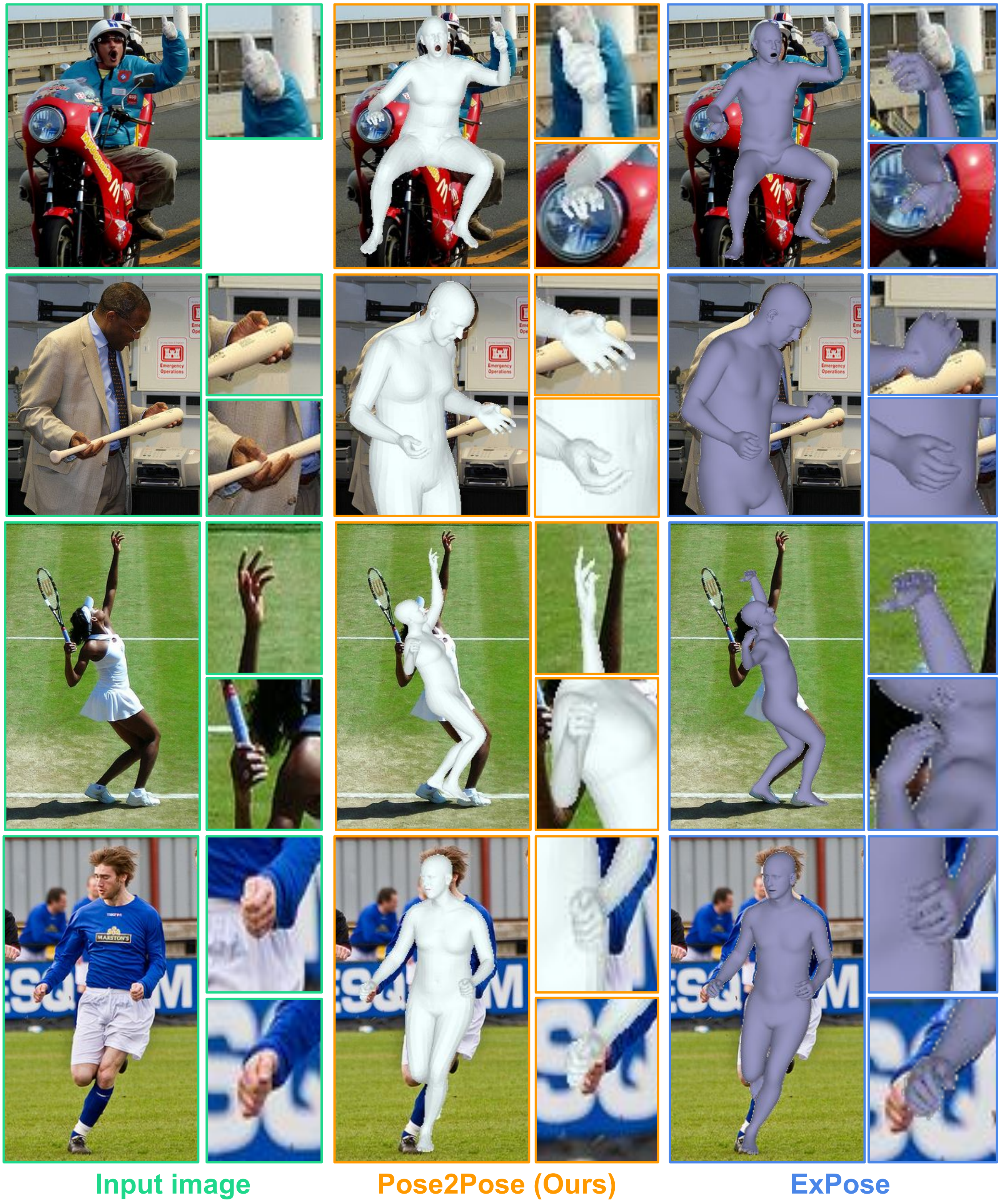

Figure 9: Qualitative comparison between Pose2Pose and ExPose [6] on MSCOCO. Hands results are zoomed.

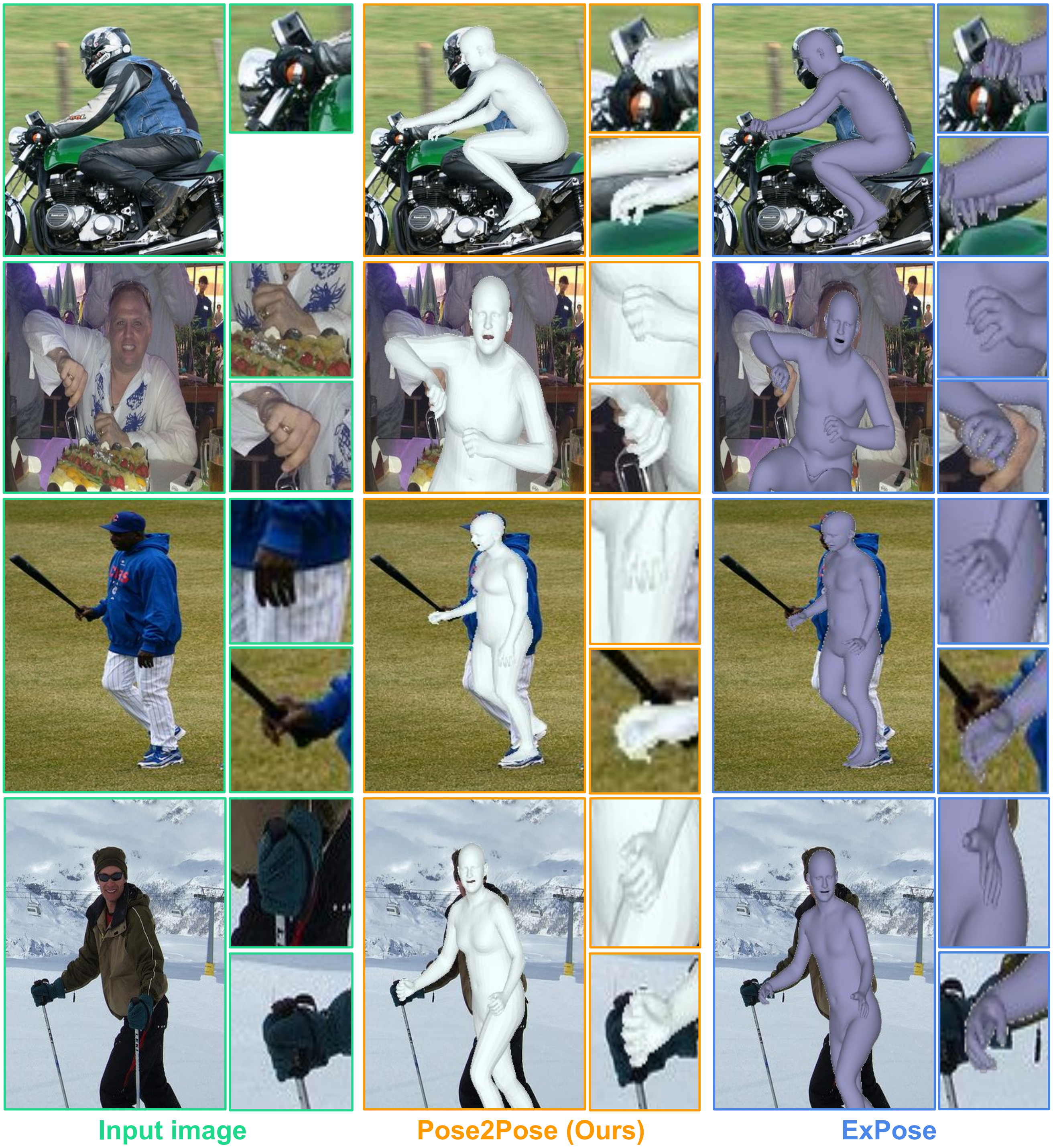

Figure 10: Qualitative comparison between Pose2Pose and ExPose [6] on MSCOCO. Hands results are zoomed.

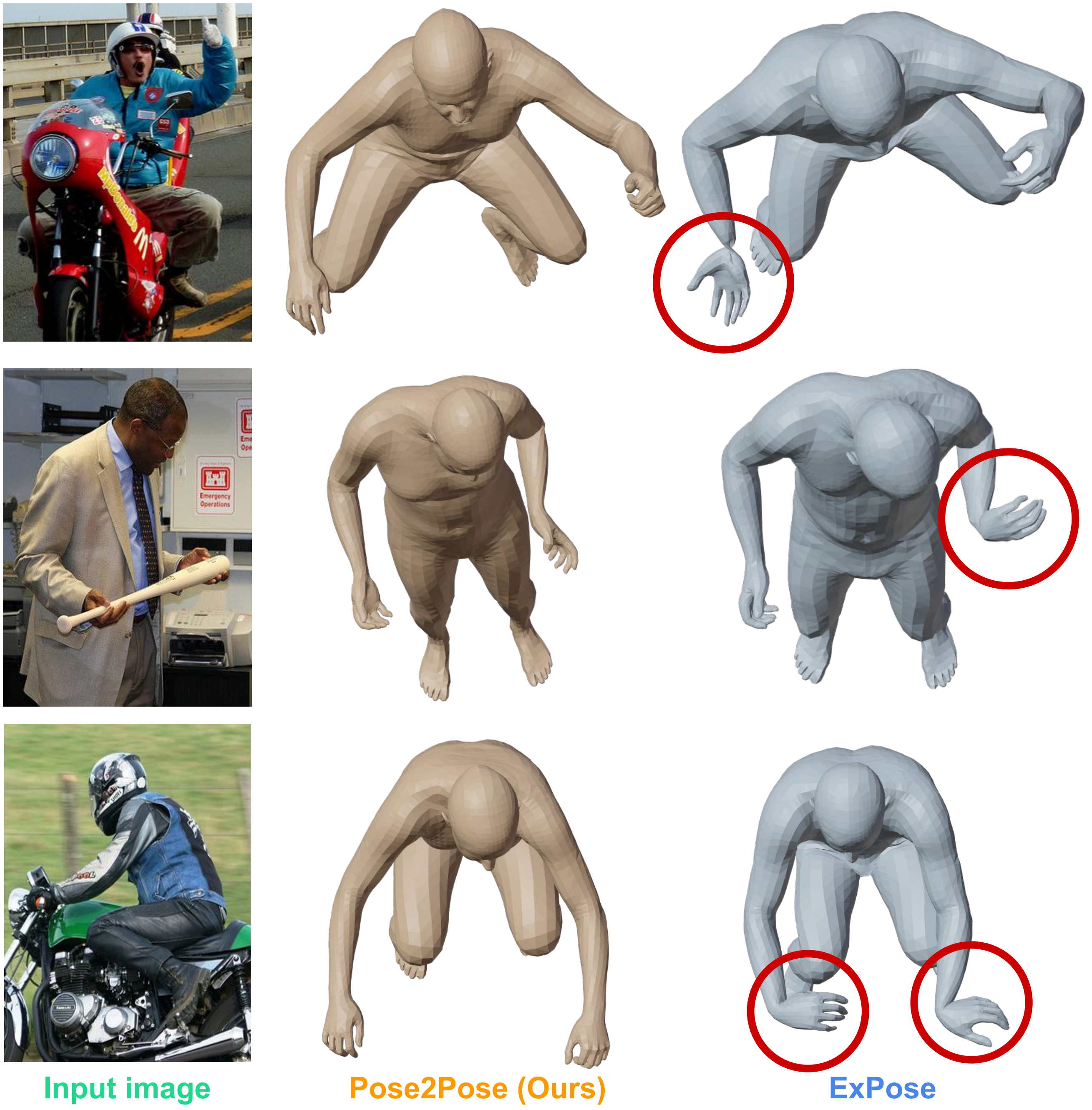

Figure 11: Qualitative comparison between Pose2Pose and ExPose [6] on MSCOCO. Implausible 3D hands results are highlighted.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 6, 10

[2] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. Pushing the envelope for RGB-based dense 3D hand pose estimation via neural rendering. In *CVPR*, 2019. 3

[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 *ECCV*, 2016. 2

[4] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3D hand shape and pose from images in the wild. In *CVPR*, 2019. 3, 10

[5] 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. 2, 3, 9, 10

[6] Vasileios Choutas, Georgios Pavlakos, Timo Bolkart, Dimitrios Tzionas, and Michael J Black. Monocular expressive body regression through body-driven attention. In *ECCV*, 2020. 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13

[7] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. RetinaFace: Single-shot multi-level face localisation in the wild. In *CVPR*, 2020. 10

[8] Zhen-Hua Feng, Patrik Huber, Josef Kittler, Peter Hancock, Xiao-Jun Wu, Qijun Zhao, Paul Koppen, and Matthias Rätsch. Evaluation of dense 3D reconstruction from 2D face images in the wild. *FG*, 2018. 10

[9] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3D hand shape and pose estimation from a single RGB image. In *CVPR*, 2019. 3

[10] Riza Alp Guler and Iasonas Kokkinos. HoloPose: Holistic 3D human reconstruction in-the-wild. In *CVPR*, 2019. 2, 4

[11] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. DensePose: Dense human pose estimation in the wild. In *CVPR*, 2018. 4

[12] 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. 10

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. 5, 10

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 4, 5, 10

[15] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *TPAMI*, 2014. 6, 10

[16] Sheng Jin, Lumin Xu, Jin Xu, Can Wang, Wentao Liu, Chen Qian, Wanli Ouyang, and Ping Luo. Whole-body human pose estimation in the wild. In *ECCV*, 2020. 4, 6, 10

[17] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3D deformation model for tracking faces, hands, and bodies. In *CVPR*, 2018. 3

[18] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 2, 6, 9

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 10

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 6

[21] 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. 2, 6, 9, 10

[22] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *CVPR*, 2019. 9

[23] Dominik Kulon, Riza Alp Guler, Iasonas Kokkinos, Michael M Bronstein, and Stefanos Zafeiriou. Weakly-supervised mesh-convolutional hand reconstruction in the wild. In *CVPR*, 2020. 3, 10

[24] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM TOG*, 2017. 2, 3

[25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 7

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 10

[27] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM TOG*, 2015. 2

[28] 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. 2, 3, 9, 10

[29] Gyeongsik Moon and Kyoung Mu Lee. NeuralAnnot: Neural annotator for in-the-wild expressive 3D human pose and mesh training sets. *arXiv preprint arXiv:2011.11232*, 2020. 6

[30] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[31] 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 *CVPR*, 2019. 2, 3, 4, 6, 7

[32] 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. 2

[33] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied Hands: Modeling and capturing hands and bodies together. *ACM TOG*, 2017. 2, 3

[34] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J Black. Learning to regress 3D face shape and expression from an image without 3D supervision. In *CVPR*, 2019. 3, 10

[35] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *ICCV*, 2017. 3

[36] Jie Song, Xu Chen, and Otmar Hilliges. Human body model fitting by learned gradient descent. In *ECCV*, 2020. 2, 9

[37] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 1

[38] Xiao Sun, Bin Xiao, Fangyin Wei, Shuang Liang, and Yichen Wei. Integral human pose regression. In *ECCV*, 2018. 4, 5

[39] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *ICCVW*, 2017. 3

[40] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3D morphable models with a very deep neural network. In *CVPR*, 2017. 3

[41] 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 *ECCV*, 2018. 10

[42] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular Total Capture: Posing face, body, and hands in the wild. In *CVPR*, 2019. 3, 7

[43] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 6

[44] 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 *CVPR*, 2020. 3

[45] Hongwen Zhang, Jie Cao, Guo Lu, Wanli Ouyang, and Zhenan Sun. Learning 3D human shape and pose from dense body parts. *TPAMI*, 2020. 4

[46] Jason Y Zhang, Sam Pepose, Hanbyul Joo, Deva Ramanan, Jitendra Malik, and Angjoo Kanazawa. Perceiving 3D human-object spatial arrangements from a single image in the wild. In *ECCV*, 2020. 8

[47] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *CVPR*, 2019. 6

[48] 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. 6, 10