PyMAF-X: Towards Well-aligned Full-body Model Regression from Monocular Images

Hongwen Zhang, Yating Tian, Yuxiang Zhang, Mengcheng Li, Liang An, Zhenan Sun, *Senior Member, IEEE*, and Yebin Liu, *Member, IEEE*

Abstract—We present PyMAF-X, a regression-based approach to recovering parametric full-body models from monocular images. This task is very challenging since minor parametric deviation may lead to noticeable misalignment between the estimated mesh and the input image. Moreover, when integrating part-specific estimations into the full-body model, existing solutions tend to either degrade the alignment or produce unnatural wrist poses. To address these issues, we propose a Pyramidal Mesh Alignment Feedback (PyMAF) loop in our regression network for well-aligned human mesh recovery and extend it as PyMAF-X for the recovery of expressive full-body models. The core idea of PyMAF is to leverage a feature pyramid and rectify the predicted parameters explicitly based on the mesh-image alignment status. Specifically, given the currently predicted parameters, mesh-aligned evidence will be extracted from finer-resolution features accordingly and fed back for parameter rectification. To enhance the alignment perception, an auxiliary dense supervision is employed to provide mesh-image correspondence guidance while spatial alignment attention is introduced to enable the awareness of the global contexts for our network. When extending PyMAF for full-body mesh recovery, an adaptive integration strategy is proposed in PyMAF-X to produce natural wrist poses while maintaining the well-aligned performance of the part-specific estimations. The efficacy of our approach is validated on several benchmark datasets for body, hand, face, and full-body mesh recovery, where PyMAF and PyMAF-X effectively improve the mesh-image alignment and achieve new state-of-the-art results. The project page with code and video results can be found at https://www.liuyebin.com/pymaf-x.

Index Terms—Expressive human mesh recovery, full-body motion capture, monocular 3D reconstruction, mesh alignment feedback.

✦

1 INTRODUCTION

Recent years have witnessed the rise of the regression-based paradigm in recovering body [1], [2], [3], [4], [5], [6], hand [7], [8], [9], [10], face [11], [12], [13], [14], and even full-body [15], [16], [17], [18] models from monocular images. These methods [1], [2], [3], [19] learn to predict model parameters directly from images in a data-driven manner. Despite the high efficiency and promising results, regression-based methods typically suffer from coarse alignment between the predicted meshes and image observations.

When recovering the parametric body or full-body models [20], [21], minor rotation errors accumulated along the kinematic chain may lead to noticeable drifts in joint positions (see the top-left example in Fig. 1), since joint poses are represented as relative rotations w.r.t. their parent joints. In order to generate well-aligned results, optimization-based methods [21], [22], [23] include data terms in the objective function so that the alignment between the projection of meshes and 2D evidence can be optimized explicitly. Similar strategies are also exploited in regression-based methods [1], [2], [3], [19] to impose 2D supervisions upon the projection of estimated meshes in the training procedure. However,

- Hongwen Zhang, Yuxiang Zhang, Mengcheng Li, Liang An, and Yebin Liu are with the Department of Automation, Tsinghua University, Beijing 100084, China. E-mail: {zhanghongwen,liuyebin}@mail.tsinghua.edu.cn; {yx-z19,li-mc18,al17}@mails.tsinghua.edu.cn; (Corresponding author: Yebin Liu)
- Yating Tian is with the Department of Computer Science and Technology, Nanjing University, Nanjing 210023, China. E-mail: yatingtian@smail.nju.edu.cn
- Zhenan Sun is with the Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China. E-mail: znsun@nlpr.ia.ac.cn.
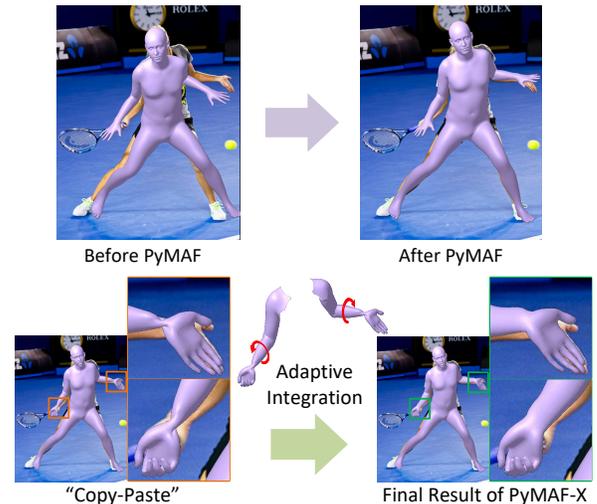

Fig. 1. Top: PyMAF improves the mesh-image alignment of the estimated mesh. Bottom: PyMAF-X produces well-aligned full-body meshes with natural wrist poses.

during testing, these deep regressors either are open-loop or simply include an Iterative Error Feedback (IEF) loop [1] in their architectures. As shown in Fig. 2(a), IEF reuses the same global feature in its feedback loop, making the regressor hardly perceive the mesh-image misalignment in the inference phase.

As suggested in previous works [24], [25], [26], [27], neural networks tend to retain high-level information and discard detailed local features when reducing the spatial size of feature maps. In order to leverage spatial information in the regression networks, it is essential to extract pixel-wise contexts for fine-grained perception. Several attempts
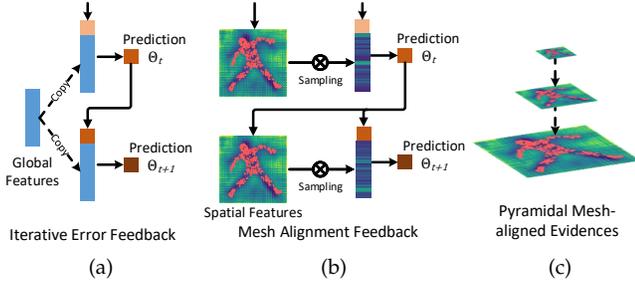
Fig. 2. (a) The commonly-used iterative error feedback. (b) The proposed mesh alignment feedback. (c) Mesh-aligned evidence extracted from a feature pyramid.

have been made to leverage pixel-wise representation such as part segmentation [28] or dense correspondences [29], [30] in their regression networks. Though pixel-level evidence is considered, it is still challenging for those methods to learn structural priors and get hold of spatial details simultaneously based merely on high-resolution contexts.

Motivated by the above observation, we design a Pyramidal Mesh Alignment Feedback (PyMAF) loop in our regression network to exploit multi-scale and position-sensitive contexts for better mesh-image alignment. The central idea of our approach is to correct parametric deviation explicitly and progressively based on the alignment status. In PyMAF, mesh-aligned evidence will be extracted from the spatial features according to the 2D projection of the estimated mesh and then fed back to the regressors for parameter updates. As illustrated in Fig. 2, the mesh alignment feedback loop takes advantage of more informative features for parameter correction compared with the commonly used iterative error feedback loop [1], [31]. In order to leverage multi-scale contexts, mesh-aligned evidence is extracted from a feature pyramid so that the coarse-aligned meshes can be corrected with large step sizes based on the lower-resolution features. To enhance these mesh-aligned features, an auxiliary task is imposed on the highest-resolution feature to infer pixel-wise dense correspondence, guiding the image encoder to preserve the most related information in the spatial feature maps. Meanwhile, a spatial alignment attention mechanism is introduced to fuse the grid and mesh-aligned features so that the regressor could be aware of the whole image contexts.

Since the SMPL family includes the hand [32] and face [33] models, PyMAF can be easily modified to reconstruct the hand and face meshes. We leverage three part-specific PyMAF networks as part experts to predict body, hand, and face parameters, and propose PyMAF-X for expressive full-body mesh recovery. Benefiting from the well-aligned results of each PyMAF-based expert, PyMAF-X can produce plausible full-body mesh results in common scenarios even using the most naive integration strategy [16]. However, as shown in Fig. 1, the naive "Copy-Paste" integration may lead to unnatural wrist poses under challenging cases. To address this issue, we propose an adaptive integration strategy to adjust the twist rotation of the elbow poses so that the elbow and wrist poses could be more compatible. In this way, the updated twist rotation of the elbow joint serves as compensation for the wrist joint and helps to produce natural wrist poses in the full-body model. Moreover, since the twist component [34] of

the elbow poses is the rotation around the elbow-to-wrist bone, it barely changes the position of the body and hand joints, which is the key to maintaining the well-aligned performances of body and hand experts. Different from existing full-body solutions [15], [16], [17], [18], our method do not rely on additional networks to infer the wrist poses, and hence bypass the learning issue raised by insufficient full-body mesh annotations.

The contributions of this work can be summarized as follows:

- A mesh alignment feedback loop is proposed for regression-based human mesh recovery, where mesh-aligned evidence is exploited to correct parametric errors explicitly so that the estimated meshes can be better aligned with the input images.
- A feature pyramid is incorporated with the mesh alignment feedback loop so that the regression network can leverage multi-scale contexts. This yields the Pyramidal Mesh Alignment Feedback (PyMAF) loop, a novel architecture for human mesh recovery.
- An auxiliary pixel-wise supervision and spacial alignment attention are introduced in PyMAF to enhance the mesh-aligned features such that they can be more informative, relevant, and aware of the whole image contexts.
- PyMAF is further extended as PyMAF-X for full-body mesh recovery, where an adaptive integration strategy with the elbow-twist compensation is proposed to avoid unnatural wrist poses while maintaining the alignment of the body and hand estimations.

An early version of this work has been published as a conference paper [6]. We have made significant extensions to our previous work [6] from three aspects. First, PyMAF is improved to be more accurate with the newly introduced spatial alignment attention, which effectively enhances the feature learning and further improves the mesh-image alignment. Second, PyMAF goes beyond body mesh recovery and is extended to reconstruct hand and full-body models from monocular images. The well-aligned performance of the body- and hand-specific PyMAF makes it more promising to produce well-aligned full-body meshes. Third, an adaptive integration strategy is proposed to assemble predictions from body and hand experts. Such a strategy effectively addresses the unnatural wrist issues while maintaining the part-specific alignment. Based on these updates, our final method PyMAF-X achieves new state-of-the-art results both qualitatively and quantitatively, contributing novel solutions towards the well-aligned and natural recovery of full-body models from monocular images.

2 RELATED WORK

2.1 Monocular Human Mesh Recovery

Monocular recovery of human meshes has been actively studied in recent years. Aiming at the same goal of producing well-aligned and natural results, two different paradigms for human mesh recovery have been investigated in the research community. In this subsection, we give a brief review of these two paradigms and refer readers to [35] for a more comprehensive survey.

Optimization-based Approaches. Pioneering work in this field mainly focus on the optimization process of fitting

parametric models (e.g., SCAPE [36] and SMPL [20]) to 2D observations such as keypoints and silhouettes [22], [37], [38]. In their objective functions, prior terms are designed to penalize the unnatural shape and pose, while data terms measure the fitting errors between the re-projection of meshes and 2D evidence. Based on this paradigm, different updates have been investigated to incorporate information such as 2D/3D body joints [22], [39], silhouettes [23], [40], part segmentation [41], dense correspondences [42] in the fitting procedure. Despite the well-aligned results obtained by these optimization-based methods, their fitting process tends to be slow and sensitive to initialization. Recently, Song et al. [43] exploit the learned gradient descent in the fitting process. Though this solution leverages rich 2D pose datasets and alleviates many issues in traditional optimization-based methods, it still relies on the accuracy of 2D poses and breaks the end-to-end learning. Alternatively, our solution supports end-to-end learning and is also able to leverage rich 2D datasets thanks to the progress (e.g., SPIN [3], EFT [44], and NeuralAnnot [45]) in the generation of more precise pseudo 3D ground-truth for 2D datasets [46], [47], [48].

Regression-based Approaches. Alternatively, taking advantage of the powerful nonlinear mapping capability of neural networks, recent regression-based approaches [1], [3], [15], [19], [28], [49], [50], [51] have made significant advances in predicting human models directly from monocular images. These deep regressors take 2D evidence as input and learn model priors implicitly in a data-driven manner under different types of supervision signals [1], [52], [53], [54], [55], [56], [57], [58] during the learning procedure. To mitigate the learning difficulty of the regressor, different network architectures have also been designed to leverage proxy representations such as silhouette [19], [59], 2D/3D joints [4], [18], [19], [42], [49], [50], [52], [60], [61], segmentation [28], [62] and dense correspondences [29], [30]. Such strategies can benefit from synthetic data [29], [63] and the progress in the estimation of proxy representations [27], [64], [65], [66], [67]. In these regressors, though supervision signals are imposed on the re-projected models to penalize the mismatched predictions during training, their architectures can hardly perceive the misalignment during the inference phase. In comparison, the proposed PyMAF is a close-loop for both training and inference, which enables a feedback loop in our regressor to leverage spatial evidence for better mesh-image alignment of the estimated human models.

Directly regressing model parameters from images is very challenging, even for neural networks. Existing methods have also offered non-parametric solutions to reconstruct human body models. Among them, volumetric representation [59], [68], mesh vertices [2], [51], [69], and position maps [70], [71], [72], [73] have been adopted as regression targets. Using non-parametric representations as the regression targets is more readily to leverage high-resolution features but needs further processing to retrieve parametric models from the outputs. Besides, the mesh surfaces of non-parametric outputs tend to be rough and more sensitive to occlusions without additional structure priors. In our solution, the deep regressor uses spatial features at multiple scales for both high-level and fine-grained perception. It produces parametric models directly with no further processing required.

Recently, there are also numerous efforts devoted to achieving or handling multi-person recovery [61], [74], [75], [76], [77], [78], [79], video inputs [80], [81], [82], [83], [84], [85], occlusions [5], [30], [86], [87], more accurate shape [63], [88], [89], ambiguities [90], [91], camera estimation [92], [93], imbalanced data [94], [95], pseudo ground-truth generation [3], [44], [45], and clothed human reconstruction [96], [97], [98], [99]. Our work is complementary to them and focuses on the design of regressor architectures for single-image well-aligned body and full-body mesh recovery.

2.2 Full-body Mesh Recovery

Compared with the large number of solutions for the body-only [1], [2], [3], [4], [30], hand-only [7], [9], [10], [100], [101], [102], [103], [104], [105], [106], [107], and face-only [11], [12], [13], [14], [108], [109] mesh recovery, the full-body mesh recovery receives less attention due to its challenging nature and the lack of annotated datasets. Similar to the developments of body-only mesh recovery algorithms, the research in the field of full-body mesh recovery begins with the proposal of full-body models, including Frank [110], Adam [110], SMPL-X [21], and GHUM [111], etc., and their corresponding optimization-based methods [21], [39], [110], [111], [112], [113]. Recently, several regression-based methods [15], [16], [17], [18], [114] have been proposed to overcome the slow and unnatural issues of optimization-based methods.

Following the pioneering work ExPose [15], regression-based methods [15], [16], [17], [18], [115], [116], [117] typically consist of three part-specific modules, namely part experts, to predict parameters of body, hand, and face from the corresponding part images cropped from original inputs. They differ mainly in the architecture of the part experts and the strategy to integrate part estimations. As the part experts are basically chosen from the body- or hand-only mesh recovery solutions, the integration strategy to sew up independent estimations becomes an essential aspect of a regression-based full-body method. The most straightforward strategy to integrate the body and hand estimations would be the "Copy-Paste" [15], [16]. To obtain more natural integration results, learning-based strategies are proposed in recent state-of-the-art methods [16], [17], [115]. For instance, FrankMocap [16] learns to correct the arm poses based on the distance between the wrist positions predicted by body and hand experts. Zhou et al. [115] incorporate body features in the learning of the hand expert so that the predicted hand poses could be more compatible with the arm. PIXIE [17] introduces a learnable moderator to merge body and hand features for the regression of wrist and finger poses. All the above solutions rely on additional networks to predict or correct the wrist poses with the condition of body information, which is typically inferior to the original hand poses predicted by the hand expert, resulting in degraded alignment on the hand parts. Recently, Hand4Whole [18] proposes to learn wrist poses based on the positions of selected hand joints but does not consider the compatibility of arm poses. In contrast to existing solutions, PyMAF-X resorts to the adjustment of the twist components [34] of wrist and elbow poses,
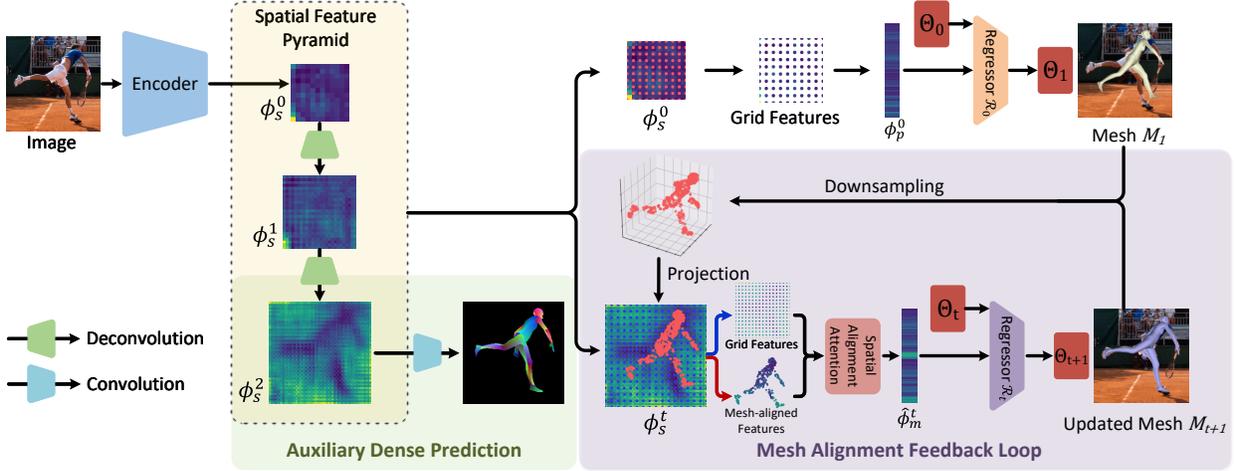
Fig. 3. Illustration of the proposed Pyramidal Mesh Alignment Feedback (PyMAF) for human mesh recovery. PyMAF leverages a feature pyramid and enables an alignment feedback loop in our network. Given a coarse-aligned model prediction, mesh-aligned evidence is extracted from finer-resolution features accordingly and fed back to a regressor for parameter rectification.

which produces natural wrist rotations while maintaining the well-aligned performances of each part expert during the integration. Besides, our motivation and method also differ from the previous work [50], [118] that decomposes the twist components in the inverse kinematics problem.

2.3 Iterative Fitting in Regression Tasks

Strategies for incorporating fitting processes along with regression tasks have also been investigated in the literature. For human mesh recovery, Kolotouros et al. [3] combine an iterative fitting procedure with the training procedure to generate more accurate ground truths for better supervision. Several attempts have been made to deform human meshes so that they can be aligned with the intermediate estimates such as depth maps [119], part segmentation [116], and dense correspondences [42]. These approaches adopt intermediate estimations as fitting objectives and hence rely on their quality. In contrast, our approach uses the currently estimated meshes to extract deep features for refinement, enabling the fully end-to-end learning of the deep regressor.

In a broader view, remarkable efforts have been made to involve iterative fitting strategies in other computer vision tasks, including facial landmark localization [120], [121], [122], human/hand pose estimation [31], [123], etc. For generic objects, Pixel2Mesh [124] progressively deforms an initial ellipsoid by leveraging perceptual features. Following the spirit of these works, we exploit new strategies to extract fine-grained evidence and contribute novel solutions in the context of human mesh recovery.

3 METHOD

In this section, we will elaborate technical details of our approach. We first present PyMAF, a powerful model for regression-based human mesh recovery, then extend it to PyMAF-X for full-body mesh recovery.

3.1 PyMAF for Body-only Mesh Recovery

As illustrated in Fig. 3, PyMAF consists of a feature pyramid for mesh recovery in a coarse-to-fine fashion. Coarse-aligned predictions will be improved by utilizing the mesh-aligned evidence extracted from spatial feature maps. In order to enhance the mesh-aligned evidence, an auxiliary dense prediction task is imposed on the image encoder while a spatial alignment attention is applied to fuse the grid and mesh-aligned features.

3.1.1 Feature Pyramid for Body Model Regression

Our image encoder aims to generate a pyramid of spatial features from coarse to fine granularities, which provide descriptions of the posed person at different scale levels. The feature pyramid will be used in subsequent predictions of the SMPL model with the pose, shape, and camera parameters $\Theta = \{\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\pi}\}$.

Formally, the encoder takes an image I as input and outputs a set of spatial features $\{\boldsymbol{\phi}_s^t \in \mathbb{R}^{C_s \times H_s^t \times W_s^t}\}_{t=0}^{T-1}$ at the end, where H_s^t and W_s^t are monotonically increasing. At level t, based on the feature map $\boldsymbol{\phi}_s^t$, a set of sampling points X^t will be used to extract point-wise features. Specifically, for each 2D point x in X^t, point-wise features $\boldsymbol{\phi}_s^t(x) \in \mathbb{R}^{C_s \times 1}$ will be extracted from $\boldsymbol{\phi}_s^t$ accordingly using the bilinear sampling. These point-wise features will go through a MLP (multi-layer perceptron) for dimension reduction and be further concatenated together as a feature vector $\boldsymbol{\phi}_p^t$, i.e.,

$$\boldsymbol{\phi}_p^t = \mathcal{F}(\boldsymbol{\phi}_s^t, X^t) = \oplus \left(\left\{ f\left(\boldsymbol{\phi}_s^t(x) \right), \text{for } x \text{ in } X^t \right\} \right), \quad (1)$$

where $\mathcal{F}(\cdot)$ denotes the feature sampling and processing operations, $\oplus$ denotes the concatenation, and $f(\cdot)$ is the MLP. After that, a parameter regressor $\mathcal{R}_t$ takes features $\boldsymbol{\phi}_p^t$ and the current estimation of parameters Θ_t as inputs and outputs the parameter residual. Parameters are then updated as Θ_{t+1} by adding the residual to Θ_t. For the level $t = 0$, Θ_0 adopts the mean parameters calculated from training data.

Given the parameter predictions Θ (the subscript t is omitted for simplicity) at each level, a mesh with vertices of $M = \mathcal{M}(\boldsymbol{\theta}, \boldsymbol{\beta}) \in \mathbb{R}^{N \times 3}$ can be generated accordingly, where $N = 6890$ denotes the number of vertices in the SMPL model. These mesh vertices are mapped to sparse 3D joints $J \in \mathbb{R}^{N_j \times 3}$ by a pretrained linear regressor, and further

projected on the image coordinate system as 2D keypoints $K = \mathbf{\Pi}(J) \in \mathbb{R}^{N_j \times 2}$, where $\mathbf{\Pi}(\cdot)$ denotes the projection function based on the camera parameters $\boldsymbol{\pi}$. Note that the pose parameters in Θ are represented as relative rotations along kinematic chains, and minor parameter errors can lead to noticeable misalignment between the 2D projection and image evidence. To penalize such misalignment during the training of the regression network, we follow common practices [1], [3] to add 2D supervisions on the 2D keypoints projected from the estimated mesh. Meanwhile, additional 3D supervisions on 3D joints and model parameters are added when ground truth 3D labels are available. Overall, the loss function for the parameter regressor is written as

$$\mathcal{L}_{reg} = \lambda_{2d}||K - \hat{K}|| + \lambda_{3d}||J - \hat{J}|| + \lambda_{para}||\Theta - \hat{\Theta}||, \quad (2)$$

where $|| \cdot ||$ is the squared L2 norm, $\hat{K}$, $\hat{J}$, and $\hat{\Theta}$ denote the ground truth 2D keypoints, 3D joints, and model parameters, respectively.

One of the improvements over the commonly used parameter regressors is that our regressors can better leverage spatial information. Unlike the commonly used regressors taking the global features $\phi_g \in \mathbb{R}^{C_g \times 1}$ as input, our regressor uses the point-wise information obtained from spatial features ϕ_s^t. A straightforward strategy to extract point-wise features would be using grid-pattern points X_{grid} and uniformly sampling features from ϕ_s^t. In the proposed approach, the sampling points X^t adopt the grid pattern at the level $t = 0$ and will be updated according to the current estimates when $t > 0$. We will show that such a mesh conditioned sampling strategy helps the regressor produce well-aligned reconstruction results.

3.1.2 Mesh Alignment Feedback Loop

As mentioned in HMR [1], directly regressing mesh parameters in one go is challenging. To tackle this issue, HMR uses an Iterative Error Feedback (IEF) loop to iteratively update Θ by taking the global features ϕ_g and the current estimation of Θ as input. Though the IEF strategy reduces parameter errors progressively, it uses the same global features each time for parameter update, which lacks fine-grained information and is not adaptive to new predictions. By contrast, we propose a Mesh Alignment Feedback (MAF) loop so that mesh-aligned evidence can be leveraged in our regressor to rectify current parameters and improve the mesh-image alignment of the estimated model.

Mesh-aligned Features. In the proposed mesh alignment feedback loop, we extract mesh-aligned features from ϕ_s^t based on the currently estimated mesh M_t when $t > 0$ to obtain more fine-grained and position-sensitive evidence. Compared with the global features or the uniformly sampled grid features, mesh-aligned features can reflect the mesh-image alignment status of the current estimation, which is more informative for parameter rectification. Specifically, the sampling points X^t are set as the mesh-aligned points X_{mesh}^t, which are obtained by first down-sampling the mesh M_t to $\tilde{M}_t$ and then projecting it on the 2D image plane, i.e., $X^t = X_{mesh}^t = \mathbf{\Pi}(\tilde{M}_t)$. Based on X_{mesh}^t, the mesh-aligned features ϕ_m^t will be extracted from ϕ_s^t using Eq. 1, i.e.,

$$\phi_m^t = \phi_p^t = \mathcal{F}(\phi_s^t, \mathbf{\Pi}(\tilde{M}_t)). \quad (3)$$
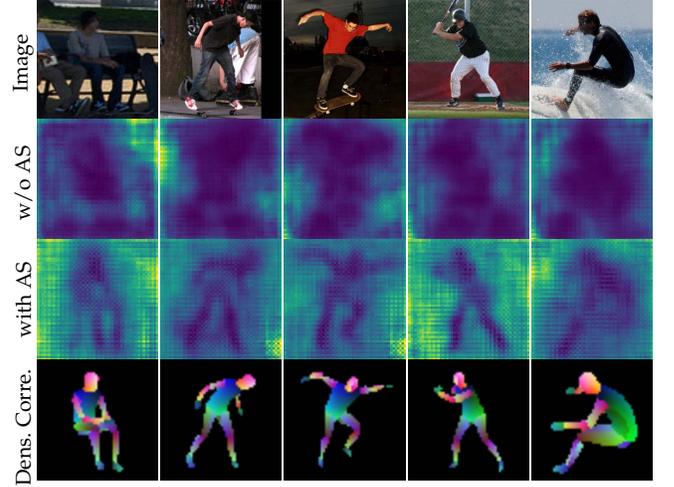

Fig. 4. Visualization of the spatial feature maps and predicted dense correspondences. Top: Input images. Second / Third Row: Spatial feature maps learned without/with Auxiliary Supervision (AS). Bottom: Predicted dense correspondence maps under auxiliary supervision.

Spatial Alignment Attention. Though the mesh-aligned features ϕ_m^t are position-sensitive, these features are confined to the re-projection regions of the current mesh result. To enable the perception of the relative positions in the whole image context, we further design spatial alignment attention to fuse the information from both grid and mesh-aligned features. Considering that both these two features are extracted from the same spatial feature map, we adopt a self-attention module to process them. Specifically, the point-wise features extracted based on the grid-pattern points X_{grid} and the mesh-aligned points X_{mesh}^t are first concatenated together as ϕ_{gm}^t:

$$\phi_{gm}^t = \left\{ \phi_s^t(x), \text{for } x \text{ in } \{X_{grid} \bigcup X_{mesh}^t\} \right\} \in \mathbb{R}^{N_{gm} \times C_s},$$
$$(4)$$

where N_{gm} is the total number of the grid-pattern and mesh-aligned points. Then, spatial alignment attention is applied to learn attentive relations among ϕ_{gm}^t so that the mesh-aligned features can be more effectively enhanced with the spatial information in the grid features. In our solution, a self-attention module [125] is employed to process the features ϕ_{gm}^t:

$$\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} = \phi_{gm}^t \boldsymbol{W}^Q, \phi_{gm}^t \boldsymbol{W}^K, \phi_{gm}^t \boldsymbol{W}^V,$$
$$\hat{\phi}_{gm}^t = Att(\boldsymbol{Q}, \boldsymbol{K})\boldsymbol{V}, \quad (5)$$

where $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$, and $\boldsymbol{W}^V$ are the learnable matrices used to generate different subspace representations of the query, key, and value features $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$, respectively, $Att(\cdot)$ denotes the scaled dot-product attention function [125] with softmax. In this way, the messages of the grid and mesh-aligned features can be fully fused together since the self-attention mechanism captures the relationships between all elements of the features ϕ_{gm}^t. After that, the enhanced mesh-aligned features $\hat{\phi}_m^t$ are obtained by reducing the dimension of $\hat{\phi}_{gm}^t$ and concatenating them together. Finally, the enhanced mesh-aligned features $\hat{\phi}_m^t$ are fed into the regressor $\mathcal{R}_t$ for parameter update:

$$\Theta_{t+1} = \Theta_t + \mathcal{R}_t \left(\Theta_t, \hat{\phi}_m^t \right), \text{for } t > 0. \quad (6)$$
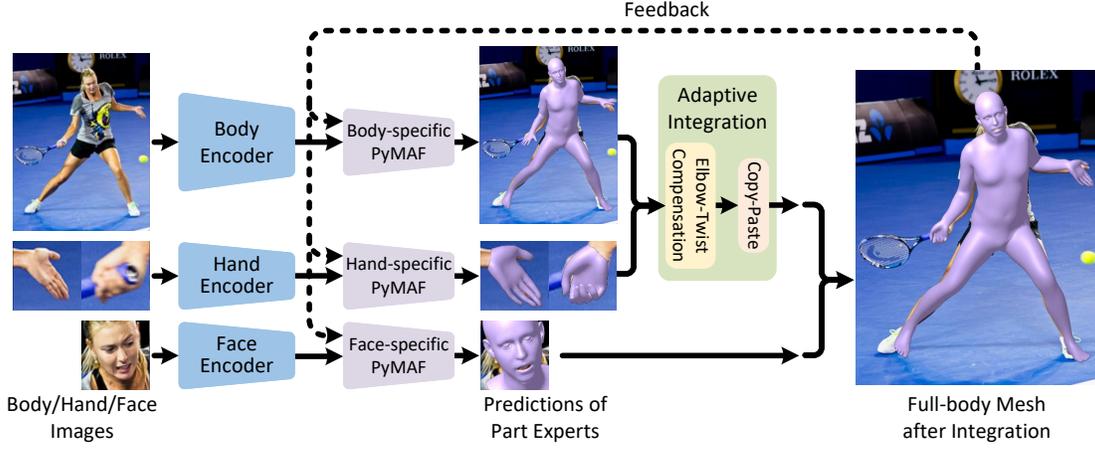
Fig. 5. The overall pipeline of PyMAF-X for full-body mesh recovery. PyMAF-X consists of three part-specific PyMAFs for part mesh prediction and integrates them together via the proposed adaptive integration strategy.

3.1.3 Auxiliary Dense Supervision

As depicted in the second row of Fig. 4, spatial features tend to be affected by noisy inputs, since raw images may contain a large amount of unrelated information such as occlusions, appearance, and illumination variations. To improve the reliability of the mesh-aligned cues extracted from spatial features, we impose an auxiliary pixel-wise prediction task on the spatial features at the last level. Specifically, during training, the spatial feature maps ϕ_s^{T-1} will go through a convolutional layer to generate dense correspondence maps with pixel-wise supervision. Dense correspondences encode the mapping relationship between foreground pixels on the 2D image plane and mesh vertices in 3D space. In this way, the auxiliary supervision provides mesh-image correspondence guidance for the image encoder to preserve the most related information in the spatial feature maps.

In our implementation, we adopt the IUV maps defined in DensePose [66] as the dense correspondence representation, which consists of the part index and UV values of the mesh vertices. Note that we do not use DensePose annotations in the dataset but render IUV maps based on the ground-truth SMPL models [30]. During training, classification and regression losses are applied on the part index P and UV channels of dense correspondence maps, respectively. Specifically, for the part index P channels, a cross-entropy loss is applied to classify a pixel belonging to either background or one among body parts. For the UV channels, a smooth L1 loss is applied to regress the corresponding UV values of the foreground pixels. Only the foreground regions are taken into account in the UV regression loss, i.e., the estimated UV channels are firstly masked by the ground-truth part index channels before applying the regression loss. Overall, the loss function for the auxiliary pixel-wise supervision is written as

$$
\begin{aligned}
\mathcal{L}_{aux} =&\lambda_{pi} CrossEntropy(P, \hat{P}) \\
&+\lambda_{uv} SmoothL1(\hat{P} \odot U, \hat{P} \odot \hat{U}) \\
&+\lambda_{uv} SmoothL1(\hat{P} \odot V, \hat{P} \odot \hat{V}),
\end{aligned} \quad (7)
$$

where $\odot$ denotes the mask operation. Note that the auxiliary prediction is required in the training phase only.

Fig. 4 visualizes the spatial features of the encoder trained with and without auxiliary supervision, where the feature maps are simply added along the channel dimension as grayscale images and visualized with colormap. We can see that the spatial features are more neat and robust to input variations after applying auxiliary supervision. Note that the dense correspondence is not limited to the IUV representation, the Projected Normalized Coordinate Code (PNCC) [126] can be also adopted as dense correspondences when IUV is not defined in the mesh model. More discussions about the choice of dense correspondences can be found in the Supplementary Material.

3.2 PyMAF-X for Full-body Mesh Recovery

The body-specific PyMAF can be easily modified to reconstruct hand and face meshes by simply changing the SMPL model in the above formulation to the MANO [32] and FLAME [33] models. Based on the regression power of PyMAF, we extend it to PyMAF-X for full-body mesh recovery.

Following previous works [15], [16], [17], [18], PyMAF-X consists of three experts, i.e., three part-specific PyMAFs, to predict the parameters of body, hand, and face, as illustrated in Fig. 5. To ensure high-resolution observations of part regions, part experts perform individual predictions on the body, hand, and face images cropped from the original inputs. At each iteration of the mesh alignment feedback loop, the predictions of the body-, hand-, and face-specific PyMAF are collected and integrated as the parameters $\Theta_{fb} = \{\boldsymbol{\theta}_{fb}, \boldsymbol{\beta}_{fb}, \boldsymbol{\psi}, \boldsymbol{\pi}\}$ of the full-body model SMPL-X [21], where $\boldsymbol{\theta}_{fb}$, $\boldsymbol{\beta}_{fb}$, and $\boldsymbol{\psi}$ denotes the pose, shape, and facial expression parameters, respectively. The pose parameters $\boldsymbol{\theta}_{fb}$ consist of the rotational poses of 55 joints in total, including 22 joints for the body, 30 finger joints for the hands, and 3 jaw joints for the face. The camera parameters $\boldsymbol{\pi}$ are taken from the predictions of the body-specific PyMAF and used to project body, hand, and face vertices on the image plane. Moreover, considering that the positions of hand and face are susceptible to inaccurate body pose estimations, we align the center of their re-projected points to the image center of hand and face to ensure their mesh-aligned features are meaningful.

Naive Integration. After individual regression of each part, we need to figure out the rotation of wrist joints to integrate the body and hand meshes. The most straightforward
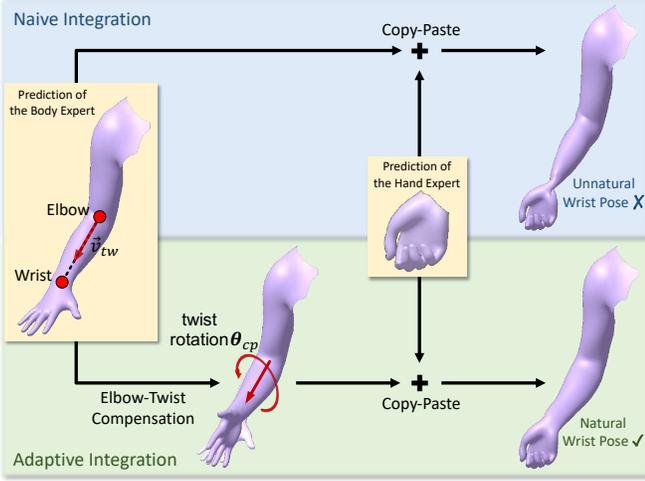
Fig. 6. Illustration of the naive integration and the proposed adaptive integration with elbow-twist compensation.

strategy would be the naive "Copy-Paste" integration [16]. Specifically, the poses of the wrist joints are calculated based on the body poses predicted by the body expert and the global orientation of hands predicted by the hand expert. Let $\acute{\boldsymbol{\theta}}_{hand}$ be the global orientation of the left or right hand, which is also the global rotation of the wrist joint. The wrist pose of the full-body model can be solved by first computing the global rotation $\acute{\boldsymbol{\theta}}_{elbow}$ of the elbow joint and then the relative rotation $\boldsymbol{\theta}_{wrist}$ of the wrist joint, i.e.,

$$\acute{\boldsymbol{\theta}}_{elbow} = \prod_{j \in A(elbow)} \boldsymbol{\theta}_j,$$
$$\boldsymbol{\theta}_{wrist} = \acute{\boldsymbol{\theta}}_{hand}^{-1} \acute{\boldsymbol{\theta}}_{elbow}, \tag{8}$$

where $\boldsymbol{\theta}_j$ denotes the relative rotation of the j-th body joint, $A(elbow)$ the ordered set of joint ancestors of the elbow joint and itself in the kinematic tree, and $\acute{\boldsymbol{\theta}}_{hand}^{-1}$ the inverse global rotation of the hand. Benefiting from the well-aligned results of each part, PyMAF-X can produce plausible results in common scenarios using such a simple integration strategy.

Adaptive Integration with Elbow-Twist Compensation. As pointed out in previous work [15], the body expert hardly perceives the hand poses due to the small proportion of hand region in the body images. It may lead to incompatible configurations of the arm and hand poses predicted individually by the body and hand experts, resulting in unnatural wrist poses of the full-body model, as illustrated in Fig. 6. Previous work [15], [16], [17], [18] alleviates this issue by learning wrist poses from the body and hand features but typically degrades the accuracy of the wrist poses and alignment. In our work, we propose an adaptive integration strategy to correct the elbow poses directly based on the solved wrist poses such that the elbow and wrist poses could be more compatible. To maintain the mesh-image alignment, we only correct the twist rotation of the elbow joints as it is the rotation along the elbow-wrist bone and barely affects the position of the body and hand joints. To this end, we first compute the twist angle of the wrist poses w.r.t. the elbow-to-wrist bone, then update the elbow and wrist poses by adding and subtracting the compensated twist rotation, respectively.

Step 1: Computing the original twist angle. The twist component around the elbow-to-wrist vector can be decomposed from the wrist poses. Without loss of generality, let the quaternion representation of the left or right wrist pose solved in Eq. (8) be $\boldsymbol{q}_{wrist} = (w_{wrist}, \vec{\boldsymbol{v}}_{wrist})$. By using Huyghe's method [127], [128], the quaternion $\boldsymbol{q}_{tw}$ of the twist rotation around the normalized elbow-to-wrist vector $\vec{\boldsymbol{v}}_{tw}$ can be calculated as:

$$u_{proj} = \frac{\vec{\boldsymbol{v}}_{wrist} \cdot \vec{\boldsymbol{v}}_{tw}}{||\vec{\boldsymbol{v}}_{wrist}||},$$
$$\boldsymbol{q}_{proj} = (w_{wrist}, u_{proj}\vec{\boldsymbol{v}}_{tw}), \tag{9}$$
$$\boldsymbol{q}_{tw} = \frac{\boldsymbol{q}_{proj}}{||\boldsymbol{q}_{proj}||},$$

where $u_{proj}\vec{\boldsymbol{v}}_{tw}$ in $\boldsymbol{q}_{proj}$ is the projection vector of the normalized $\vec{\boldsymbol{v}}_{wrist}$ onto $\vec{\boldsymbol{v}}_{tw}$. Let w_{tw} be the first element of the twist quaternion $\boldsymbol{q}_{tw}$, then the twist rotation angle can be computed as $\alpha_{tw} = 2cos^{-1}(w_{tw}) \in [-\pi, \pi]$.

Step 2: Updating elbow and wrist poses. The angle α_{tw} reflects the intensity of the wrist rotation around the elbow-to-wrist bone, and an out-of-range twist angle typically leads to unnatural wrist poses. To tackle this issue, an additional twist rotation is added to the elbow pose and serves as an compensation to the wrist pose. Specifically, the elbow/wrist poses are updated as $\tilde{\boldsymbol{\theta}}_{elbow}/\tilde{\boldsymbol{\theta}}_{wrist}$ by adding/subtracting a twist rotation $\boldsymbol{\theta}_{cp}$ around the elbow-to-wrist vector $\vec{\boldsymbol{v}}_{tw}$ with a compensation angle of α_{cp}, i.e., $\tilde{\boldsymbol{\theta}}_{elbow} = \boldsymbol{\theta}_{elbow}\boldsymbol{\theta}_{cp}$ and $\tilde{\boldsymbol{\theta}}_{wrist} = \boldsymbol{\theta}_{cp}^{-1}\boldsymbol{\theta}_{wrist}$. In our solution, we empirically set a range $[\alpha_{tmin}, \alpha_{tmax}]$ to constraint α_{tw} and adopt the compensation angle α_{cp} as:

$$\alpha_{cp} = \begin{cases} \alpha_{tw} - \alpha_{tmax}, & \text{if } \alpha_{tw} > \alpha_{tmax}, \\ min(\alpha_{tw} - \alpha_{tmin}, 0), & \text{otherwise.} \end{cases} \tag{10}$$

As shown in our experiments, with the twist compensation from the elbow joint, the wrist pose becomes more natural while maintaining the mesh-image alignment of the body and hands. In practice, the adaptive integration is not applied for those invisible hands since the global orientation predicted by the hand expert is not reliable when the hand is invisible. In our implementation, the hand expert of PyMAF-X also predicts the confidence of the visibility status of hands. When the hand is invisible, the full-body model simply adopts the wrist poses predicted by the body expert and the mean poses of hands.

4 EXPERIMENTS

4.1 Implementation Details

The part-specific PyMAF primarily adopts ResNet-50 [129] as the backbone of the image encoder. We also follow ExPose [15] and PIXIE [17] to adopt HRNet-W48 [27] as the encoder backbone for the body model regression. For each part-specific PyMAF, the image encoder takes a 224×224 image as input and produces spatial feature maps with resolutions of $\{14 \times 14, 28 \times 28, 56 \times 56\}$. When generating mesh-aligned features, the vertex number of body, hand, and face meshes is down-sampled to 431, 195, and 708, respectively. The mesh-aligned features extracted from feature maps of each point will be processed by MLPs so that their

channel dimensions will be reduced to 5. Hence, the mesh-aligned feature vector for the body model has a length of $2155 = 431 \times 5$, which is similar to the length of the global features used in HMR [1]. The maximum number T is set to 3, which is equal to the iteration number used in HMR. For the grid features used at $t = 0$, they are uniformly sampled from ϕ_s^0 with a 21×21 grid pattern, i.e., the point number is $441 = 21 \times 21$ which is approximate to the vertices number 431 after mesh downsampling. The regressors $\mathcal{R}_t$ have the same architecture as the one in HMR, except that they have slightly different input dimensions. The twist angle constraint $[\alpha_{tmin}, \alpha_{tmax}]$ is empirically set to $[-72°, 72°]$ in our implementation. Following previous work [3], [5], we adopt the continuous 6D representation [130] for pose parameters in the regressor. Following PARE [5], the body encoder is initialized with the model pretrained on 2D pose datasets [46], [48]. During training, we use the Adam [131] optimizer and set the learning rate to $5e-5$ without decay. The part-specific PyMAFs are first pre-trained individually and then assembled together for finetuning on full-body datasets. Similar to PARE [5], we also observed a slight performance gain when removing the auxiliary supervision at the final stage of training, but we do not apply such a strategy in our experiments for more consistent ablation studies of our newly introduced components. More details of the implementation can be found in our code and the Supplementary Material.

Camera Setting. We follow previous work [3] to use a weak perspective camera with a pre-defined focal length of 5,000 by default for training and evaluation. When running experiments on AGORA [132], we use a perspective camera with the focal lengths estimated by SPEC [93] as there are stronger perspective distortions in this dataset. Incorporating the camera setting of SPEC [93] with our method for more accurate mesh recovery is left for future work.

Runtime. The PyTorch implementation of the body-only PyMAF takes about 22 ms to process one sample on the machine with an NVIDIA RTX 3090 GPU. For full-body mesh recovery, PyMAF-X takes about 80 ms to process one sample, which is on par with existing regression-based approaches [15], [17], [18]. In our current implementation, the part-specific backbone networks run in sequence to process the body, hand, and face images. Running them in parallel would further reduce the runtime.

4.2 Datasets

Following the practices of previous work [1], [3], [5], the body expert is trained on a mixture of data from several datasets with 3D and 2D annotations, including Human3.6M [134], MPI-INF-3DHP [139], MPII [46], LSP [47], LSP-Extended [140], and COCO [48]. For the hand expert, we use images from FreiHAND [8], InterHand2.6M [9] and COCO-Wholebody [141] for training. For the face expert, we use the images from VGGFace2 [142] for training. Detailed descriptions of the datasets can be found in the Supplementary Material.

Pseudo Ground-truth. Following previous work [5], the SMPL/SMPL-X models fitted in EFT [44] and ExPose [15] are used as pseudo ground-truth annotations for the training of body and full-body model regressors. For the

Table 1
Reconstruction errors on 3DPW [133] and Human3.6M [134]. Backbone architectures are highlighted in the brackets.

	Method	3DPW			Human3.6M	
		PVE	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
Temporal	Kanazawa et al. [135]	139.3	116.5	72.6	-	56.9
	Doersch et al. [55]	-	-	74.7	-	-
	Arnab et al. [136]	-	-	72.2	77.8	54.3
	DSD [137]	-	-	69.5	59.1	42.4
	VIBE [80]	113.4	93.5	56.5	65.9	41.5
	MEVA [138]	-	86.9	54.7	-	-
	TCMR [81]	111.5	95.0	55.8	62.3	41.1
Multiple Stage	Pavlakos et al. [19]	-	-	-	-	75.9
	NBF [28]	-	-	-	-	59.9
	Zanfir et al. [56]	-	90.0	57.1	-	-
	I2L-MeshNet [4]	-	93.2	58.6	55.7	41.1
	Pose2Mesh [49]	-	88.9	58.3	64.9	46.3
	LearnedGD [43]	-	-	55.9	-	56.4
	HUND [116]	-	81.4	57.5	69.5	52.6
	HybrIK [50]	94.5	80.0	48.8	54.4	**34.5**
	Pose2Pose [18]	-	86.6	54.4	-	-
	3DCrowdNet [61]	98.3	81.7	51.5	-	-
Single Stage	HMR (Res50) [1]	-	130.0	76.7	88.0	56.8
	GraphCMR (Res50) [2]	-	-	70.2	-	50.1
	SPIN (Res50) [3]	116.4	96.9	59.2	62.5	41.1
	HMR-EFT (Res50) [44]	-	-	54.3	-	46.0
	Graphormer (HR64) [69]	-	-	-	**51.2**	**34.5**
	ROMP (Res50) [77]	105.6	89.3	53.5	-	-
	PARE (Res50) [5]	99.7	82.9	52.3	-	-
	PARE (HR32) [5]	97.9	82.0	50.9	-	-
	Baseline (Res50)	99.8	84.4	51.3	63.6	44.7
	PyMAF (Res50)	94.4	79.7	49.0	58.1	40.2
	Baseline (HR48)	93.3	79.5	48.0	58.8	39.5
	PyMAF (HR48)	**91.3**	78.0	**47.1**	54.2	37.2
	* with training on 3DPW					
	* HMR-EFT (Res50) [44]	-	-	51.6	-	-
	* ROMP (Res50) [77]	94.7	79.7	49.7	-	-
	* Graphormer (HR64) [69]	87.7	74.7	45.6	-	-
	* PARE (HR32) [5]	88.6	74.5	46.5	-	-
	* PyMAF (Res50)	88.7	76.8	46.8	-	-
	* PyMAF (HR48)	**87.0**	**74.2**	**45.3**	-	-

training of the face expert, we use DECA [14] and a face alignment algorithm FAN [143] to generate pseudo ground-truth FLAME models and facial landmarks on the training set of VGGFace2 [142].

Dense Correspondence. Note that we do not use the DensePose annotations in COCO for auxiliary supervision but render dense correspondence maps based on the pseudo ground-truth meshes using the method described in [30].

4.3 Evaluation Metrics

We report the results of our approach in various evaluation metrics for quantitative comparisons with existing state-of-the-art methods, where all metrics are computed in the same way as previous work [1], [3], [5], [15], [17], [18] in literature.

To quantitatively evaluate the performance of the 3D pose estimation, PVE, MPJPE, PA-PVE, and PA-MPJPE are adopted as the primary evaluation metrics. They are all reported in millimeters (mm) by default. Among these metrics, PVE denotes the mean Per-vertex Error, defined as the average point-to-point Euclidean distance between the predicted and ground truth mesh vertices, while MPJPE denotes the Mean Per Joint Position Error. PA-PVE and PA-MPJPE denote the PVE and MPJPE after rigid alignment of the prediction with the ground truth using Procrustes

Analysis. Note that the metrics PA-PVE and PA-MPJPE are not aware of the global rotation and scale errors since they are calculated after rigid alignment.

4.4 Comparison with the State of the Art

4.4.1 Evaluation on Body-only Mesh Recovery

3D Human Pose and Shape Estimation. We first evaluate our approach on the 3D human pose and shape estimation task and make comparisons with previous state-of-the-art regression-based methods. We present evaluation results for quantitative comparison on 3DPW [133] and Human3.6M [134] datasets in Table 1. Our PyMAF achieves competitive or superior results among previous approaches, including frame-based and temporal approaches. Note that the approaches reported in Table 1 are not strictly comparable since they may use different training data, pseudo ground-truths, learning rate schedules, training epochs, etc. For a fair comparison, we report our baseline results in Table 1, which is trained under the same setting as PyMAF. The baseline approach has the same network architecture with HMR [1] and also adopts the 6D rotation representation [130] for pose parameters. Under the setting of using ResNet-50 backbone and without training on 3DPW, PyMAF reduces the MPJPE over the baseline by 4.7 mm and 5.5 mm on 3DPW and Human3.6M datasets, respectively.

From Table 1, we can see that PyMAF has more notable improvements on the metrics MPJPE and PVE. We would argue that the metric PA-MPJPE can not reveal the mesh-image alignment performance since it is calculated as the MPJPE after rigid alignment. As depicted in the Supplementary Material, a reconstruction result with a smaller PA-MPJPE value can have a larger MPJPE and worse alignment between the reprojected mesh and the input image.

2D Human Pose Estimation. We evaluate 2D human pose estimation performance on the COCO validation set to measure the mesh-image alignment quantitatively in real-world scenarios. During the evaluation, we project keypoints from the estimated mesh on the image plane and compute the Average Precision (AP) based on the keypoint similarity with the ground truth 2D keypoints. The results of keypoint localization APs are reported in Table 2. OpenPose [65], a widely-used 2D human pose estimation algorithm, is also included for reference. We can see that the COCO dataset is very challenging for approaches to human mesh recovery as they typically have much worse performances in terms of the 2D keypoint localization accuracy. In Table 2, we also include the results of the optimization-based SMPLify [22] by fitting the SMPL model to the ground-truth 2D keypoints with 1,500 optimization iterations. As pointed out in previous work [3], SMPLify may produce well-aligned but unnatural results. Moreover, SMPLify is much more time-consuming than regression-based solutions. Among approaches to recovering 3D human mesh, PyMAF outperforms previous regression-based methods by remarkable margins, making it the most competitive mesh recovery method in comparison with OpenPose [65]. Under the backbone of ResNet-50, PyMAF brings significant improvements over our baseline by 8.5% and 6.2% on AP and AP_{50}, respectively. Qualitative comparisons can be found in the supplementary material.

Table 2
Keypoint localization APs on the COCO [48] validation set. Results of SMPLify [22] are evaluated based on the implementation in SPIN [3]. Results of HMR [1], GraphCMR [2], and SPIN [3] are evaluated based on their publicly released code and models.

Method	AP	AP_{50}	AP_{75}	AP_M	AP_L
OpenPose [65]	**65.3**	**85.2**	**71.3**	**62.2**	**70.7**
SMPLify [22]	29.0	47.5	30.0	35.6	24.3
HMR [1]	18.9	47.5	11.7	21.5	17.0
CMR [2]	9.3	26.9	4.2	11.3	8.1
SPIN [3]	17.3	39.1	13.5	19.0	16.6
DaNet [30]	33.8	68.6	29.9	36.0	32.3
Baseline (Res50)	33.7	65.8	31.5	36.3	31.9
PyMAF (Res50)	42.2	72.0	44.6	44.4	41.0
Baseline (HR48)	44.9	74.8	48.4	48.1	42.8
PyMAF (HR48)	**47.7**	**76.7**	**52.5**	**50.5**	**46.1**

Fig. 7. Qualitative comparison of face reconstruction results on in-the-wild images. The results of both DECA [14] and PyMAF are the parametric FLAME [33] models.

4.4.2 Evaluation on Hand-only Reconstruction

We compare the hand-only PyMAF with state-of-the-art approaches on the FreiHAND [8] dataset. As shown in Table 3, PyMAF outperforms the baseline and previous full-body methods and is comparable with recent hand-only methods [4], [51]. It is also worth noting that full-body methods typically adopt the parametric representation of the hand mesh, which tends to be numerically inferior to the non-parametric representation used in recent hand-only methods [4], [51], as pointed out in previous works [4], [35].

4.4.3 Evaluation on Face-only Reconstruction

Following previous work [15], [17], we compare the face-only PyMAF with state-of-the-art face reconstruction approaches on the test set of Stirling3D [144] and NoW [13] datasets. Table 4 reports the performances of different methods in Point-to-Surface after Procrustes Alignment (PA-P2S). It shows that the PyMAF outperforms the face expert of previous full-body methods ExPose [15] and PIXIE [17], while achieving similar results compared with the strong face-only method DECA [14]. Qualitative comparisons of face reconstruction results are visualized in Fig. 7. For more consistent comparisons, we only show the intermediate parametric FLAME model predicted by DECA [14] without using detail displacements. We can see that PyMAF is able to capture expressive face shapes and has competitive results against DECA [14].

4.4.4 Evaluation on Full-body Mesh Recovery

Following previous work [15], [16], [17], [18] on full-body mesh recovery, we evaluate the performance of

Fig. 8. Qualitative comparison of full-body model reconstruction results on the COCO [48] validation set.

Table 3
Hand reconstruction errors on the FreiHAND [8] dataset. † denotes the methods using extra training data more than FreiHAND.

Method	PA-PVE ↓	PA-MPJPE ↓	F-Scores ↑ @ 5 mm / 10 mm
* Hand-only methods			
FreiHAND [8]	10.7	-	0.529 / 0.935
Pose2Mesh [49]	7.8	7.7	0.674 / 0.969
I2L-MeshNet [4]	7.6	7.4	0.681 / 0.973
METRO [51]	**6.7**	**6.8**	**0.717 / 0.981**
* Full-body methods			
ExPose [15]	11.8	12.2	0.484 / 0.918
Zhou et al. [115]	-	15.7	- / -
FrankMocap [16]	11.6	9.2	0.553 / 0.951
PIXIE [17] †	12.1	12.0	0.468 / 0.919
Hand4Whole [18] †	7.7	**7.7**	0.664 / 0.971
Baseline	8.6	8.9	0.605 / 0.963
PyMAF	8.1	8.4	0.638 / 0.969
PyMAF †	**7.5**	**7.7**	**0.671 / 0.974**

Table 4
Face reconstruction errors on Stirling3D [144] and NoW [13] datasets.

Method	PA-P2S (mm) ↓		
	Median	Mean	Std
* Stirling3D LQ/HQ			
RingNet [13]	1.63/1.58	2.08/2.02	1.79/1.69
ExPose [15]	1.76/1.91	2.27/2.42	1.97/2.03
Baseline	1.55/1.57	2.02/2.04	1.77/1.78
PyMAF	**1.51/1.48**	**1.97/1.92**	**1.72/1.67**
* NoW			
RingNet [13]	1.21	1.54	1.31
DECA [14]	**1.09**	**1.38**	**1.18**
ExPose [15]	1.26	1.57	1.32
PIXIE [17]	1.18	1.49	1.25
Baseline	1.17	1.47	1.23
PyMAF	1.13	1.42	1.20

PyMAF-X on two benchmark datasets, i.e., EHF [21] and AGORA [132].

Table 5 reports the results of different methods for full-body mesh recovery, including the optimization-based MTC [112] and SMPLify-X [21], and the regression-based ExPose [15], FrankMocap [16], Zhou et al. [115], PIXIE [17], and Hand4Whole [18]. We can see that PyMAF-X achieves the best results among existing solutions on most metrics, especially on the evaluation of the body and full-body reconstruction.

Table 6 compares the results of PyMAF-X and other full-body methods on the test set of AGORA [132], where all the evaluation results are taken from the official evaluation platform. Recent state-of-the-art approaches to body-only mesh recovery are also included in Table 6 for comprehensive comparisons. Note that the evaluation on AGORA is

also affected by the detection results as the predictions are first matched with the ground truth and then used to calculate the reconstruction error. We use an off-the-shelf tool OpenPifPaf [145] to detect persons and the corresponding hands and face regions, of which the person detection result is slightly worse than the recent solutions Hand4Whole [18] and BEV [78]. For matched predictions, PyMAF-X outperforms other methods, especially in the metrics for hand and full-body reconstruction on this challenging dataset.

Qualitative comparisons of different full-body methods on real-world images are shown in Fig. 8, where we can see that PyMAF-X produces more accurate body, hand, and wrist poses than recent state-of-the-art approaches, including FrankMocap [16], PIXIE [17], and Hand4Whole [18]. The video results of PyMAF-X and other full-body methods can be found on our project page and supplementary materials.

Table 5
Reconstruction errors on the EHF [21] test set. † denotes the optimization-based approaches.

Method	PVE			PA-PVE				PA-MPJPE	
	Full-body	Hands	Face	Full-body	Body	Hands	Face	Body	Hands
MTC [112] †	-	-	-	67.2	-	-	-	107.8	16.7
SMPLify-X [21] †	-	-	-	65.3	75.4	12.3	6.3	87.6	12.9
ExPose [15]	77.1	51.6	35.0	54.5	52.6	12.8	5.8	62.8	13.1
FrankMocap [16]	107.6	42.8	-	57.5	52.7	12.6	-	62.3	12.9
Zhou et al. [115]	90.8	51.7	28.1	66.2	60.3	14.6	7.0	70.9	14.3
PIXIE [17]	89.2	42.8	32.7	55.0	53.0	11.1	**4.6**	61.5	11.6
Hand4Whole [18]	76.8	39.8	26.1	50.3	-	10.8	5.8	60.4	10.8
PyMAF-X (Res50)	68.0	29.8	20.5	**47.3**	45.9	**10.1**	5.6	52.9	**10.3**
PyMAF-X (HR48)	**64.9**	**29.7**	**19.7**	50.2	**44.8**	10.2	5.5	**52.8**	**10.3**

Table 6
Reconstruction errors on the AGORA test set. † denotes the methods that are fine-tuned on the AGORA training set or similarly synthetic data [93]. All results are taken from the official evaluation platform[1].

Method	Body Model	Detection	MVE ↓				MPJPE ↓			
		F1 Score ↑	Full-Body	Body	Face	L/R-Hand	Full-Body	Body	Face	L/R-Hand
SPIN [3] †	SMPL	0.77	-	148.9	-	-	-	153.4	-	-
PARE [5] †	SMPL	0.84	-	140.9	-	-	-	146.2	-	-
SPEC [93] †	SMPL	0.84	-	106.5	-	-	-	112.3	-	-
ROMP [77] †	SMPL	0.91	-	103.4	-	-	-	108.1	-	-
BEV [78] †	SMPL	0.93	-	100.7	-	-	-	105.3	-	-
SMPLify-X [21]	SMPL-X	0.71	236.5	187.0	48.9	48.3/51.4	231.8	182.1	52.9	46.5/49.6
ExPose [15]	SMPL-X	0.82	217.3	151.5	51.1	74.9/71.3	215.9	150.4	55.2	72.5/68.8
FrankMocap [16]	SMPL-X	0.80	-	168.3	-	54.7/55.7	-	165.2	-	52.3/53.1
PIXIE [17]	SMPL-X	0.82	191.8	142.2	50.2	49.5/49.0	189.3	140.3	54.5	46.4/46.0
Hand4Whole [18] †	SMPL-X	**0.94**	135.5	90.2	41.6	46.3/48.1	132.6	87.1	46.1	44.3/46.2
PyMAF-X (Res50) †	SMPL-X	0.89	134.4	90.4	38.7	45.9/47.3	132.8	89.0	42.2	43.9/45.4
PyMAF-X (HR48) †	SMPL-X	0.89	**125.7**	**84.0**	**35.0**	**44.6/45.6**	**124.6**	**83.2**	**37.9**	**42.5/43.7**

4.5 Ablation Study

In this part, we will perform ablation studies under various settings to validate the key components proposed in PyMAF and PyMAF-X. The efficacy of the mesh-aligned features, pyramidal design, auxiliary dense supervision, and spatial alignment attention proposed in PyMAF will be validated on Human3.6M [134]. As the Human3.6M dataset includes large-scale amounts of images and the corresponding ground-truth 3D labels, ablation approaches of PyMAF are trained and evaluated on Human3.6M. As for the proposed adaptive integration in PyMAF-X, ablation approaches are evaluated on EHF [21], where different approaches are trained under the same setting.

Efficacy of Mesh-aligned Features. In PyMAF, mesh-aligned features provide the current mesh-image alignment information in the feedback loop, which is essential for better mesh recovery. To verify this, we alternatively replace the mesh-aligned features with the global features or the grid features uniformly sampled from spatial features as the input for parameter regressors. Table 7 reports the performance of approaches using different types of features in the feedback loop. The results under the non-pyramidal setting are also included in Table 7, where the grid and mesh-aligned features are extracted from the feature maps with the highest resolution (i.e., 56×56), and the mesh-aligned features are extracted on the reprojected points of the mesh under the mean pose at $t = 0$. Note that all approaches in Table 7 do not use auxiliary supervision.

Unsurprisingly, using mesh-aligned features yields the

Table 7
Ablation study on using different types of feedback features for refinement. No auxiliary supervision is applied.

Feedback Feat.	Pyramid?	MPJPE	PA-MPJPE
Global (Baseline)		84.1	55.6
Grid	No	80.5	54.7
Mesh-aligned		**79.6**	**53.4**
Grid	Yes	79.7	54.3
Mesh-aligned		**76.8**	**50.9**

best performance under both non-pyramidal and pyramidal designs. The approach using the grid features sampled from spatial feature maps has better results than global features but is worse than the mesh-aligned counterpart. The mesh-aligned solution achieves even more performance gain when using pyramidal feature maps since multi-scale mesh-alignment evidence is leveraged in the feedback loop. Though the grid features contain primary spatial cues on uniformly distributed pixel positions, they can not reflect the alignment status of the current estimation. This implies that mesh-aligned features are the most informative ones for the regressor to rectify the current mesh parameters.

Benefit from Auxiliary Supervision. The auxiliary pixel-wise supervision helps to enhance the reliability of the mesh-aligned evidence extracted from spatial features. Using alternative pixel-wise supervision such as part segmentation rather than dense correspondences is also possible in our framework. In our approach, these auxiliary predictions

1. https://agora-evaluation.is.tuebingen.mpg.de

Table 8
Ablation study on using different auxiliary supervision settings and input types for regressors.

Aux. Supv.	Input Type	MPJPE	PA-MPJPE
None	Feature	76.8	50.9
Part. Seg.	Part. Seg.	108.1	75.9
	Feature	75.5	49.2
	Feature*Part. Seg.	77.6	51.1
Dense Corr.	Dense Corr.	77.8	54.7
	Feature	**75.1**	**48.9**

Table 9
Ablation study on the spatial alignment attention.

Feedback Feat.	PVE	MPJPE	PA-MPJPE
Mesh-aligned	89.1	76.8	50.9
+ Grid	88.9	76.6	51.0
+ SAA	**85.1**	**73.6**	**50.1**

are solely needed for supervision during training since the point-wise features are extracted from feature maps. For more in-depth analyses, we have also tried extracting point-wise features from the auxiliary predictions, i.e., the input type of regressors are intermediate representations such as part segmentation or dense correspondences. Table 8 compares different auxiliary supervision settings and input types for regressors during training. Using part segmentation is slightly worse than our dense correspondence solution. Compared with the part segmentation, the dense correspondences preserve clean and rich information in foreground regions. Moreover, using feature maps for point-wise feature extraction consistently performs better than auxiliary predictions. This can be explained by the fact that using intermediate representations as input for regressors hampers the end-to-end learning of the whole network. Under the auxiliary supervision strategy, the spatial feature maps are learned with the signal backpropagated from both auxiliary prediction and parameter correction tasks. In this way, the background features can also contain information for mesh parameter correction since the deep features have larger receptive fields and are trained in an end-to-end manner. As shown in Table 8, when the mesh-aligned features are masked with the foreground region of part segmentation predictions, the performance degrades from 75.5 mm to 77.6 mm on MPJPE.

Efficacy of Spatial Alignment Attention. In our approach, Spatial Alignment Attention (SAA) is designed to enable the awareness of the whole image context in the regressor. To validate its efficacy, we replace the spatial alignment attention with fully-connected layers to fuse the grid and mesh-aligned features. As reported in Table 9, simply fusing the grid features (the second row) only brings marginal improvements in comparison with the approach using the spatial alignment attention (the third row). The performances of PyMAF with or without spatial alignment attention across each refinement iteration are reported in Table 10, where the PyMAF with spatial alignment attention improves the reconstruction results more quickly.

Efficacy of Adaptive Integration. In PyMAF-X, an elbow-twist compensation is used to adaptively correct the elbow poses in the integration of body and hand estimations. Such an adaptive integration strategy could produce

Table 10
Reconstruction errors across different iterations in the feedback loop. SAA denotes the spatial alignment attention.

Method	Metric	M_0	M_1	M_2	M_3
PyMAF w /o SAA	PVE	312.1	97.0	90.5	88.7
	MPJPE	274.0	80.3	76.6	75.1
	PA-MPJPE	131.7	52.1	49.9	48.9
PyMAF w. SAA	PVE	312.1	91.2	83.6	**81.8**
	MPJPE	274.0	78.2	73.2	**72.1**
	PA-MPJPE	131.7	51.6	49.8	**48.7**

Fig. 9. Visual comparison of different integration strategies under the cases of challenging wrist poses.

physically-plausible wrist poses while preserving the mesh-image alignment. We investigate different integration strategies and compare our solution with two alternatives: i) a learned integration strategy similar to PIXIE [17], which predicts the wrist poses based on the fused features of body and hand features; ii) the naive copy-paste integration strategy [16], which calculates the wrist poses based on the estimated body and hand poses. Table 11 reports the performances of the three different integration strategies on the EHF dataset. Here, we use the MPJPE of body and hand joints to measure the mesh-image alignment and the PA-PVE of wrist vertices to measure the physical plausibility of the wrist joint. As shown in the first row, the learned integration strategy can also produce natural wrist poses but degrade the alignment of hand parts in the full-body model. Compared with the learned and copy-paste strategies, the proposed adaptive integration produces both well-aligned and natural poses of the body, hand, and wrist parts. Fig. 9 provides a visual comparison of different integration strategies under challenging cases in real-world scenarios. We can see that our adaptive integration maintains the alignment and effectively improves the plausibility of the wrist poses by leveraging the twist compensation from elbow joints.

5 CONCLUSION

In this paper, we first present Pyramidal Mesh Alignment Feedback (PyMAF) for regression-based human mesh recovery and further extend it as PyMAF-X for full-body mesh recovery. PyMAF is primarily motivated by the observation of the reprojection misalignment between the parametric

Table 11
Ablation study on the usage of the learned, copy-paste, and the proposed adaptive integration strategies.

| Integration | MPJPE (Alignment) | | PA-PVE (Plausibility) |
	Body	Hands	Wrist
Learned	**64.9**	42.3	**5.8**
Copy-Paste	**64.9**	**31.2**	6.7
Adaptive (Ours)	**64.9**	**31.2**	5.9

mesh results and the input images. At the core of PyMAF, the parameter regressor leverages spatial information from a feature pyramid to correct the parameter deviation explicitly in a feedback loop based on the alignment status of the currently estimated meshes. To achieve this, given a coarse-aligned mesh estimation, the mesh-aligned features are first extracted from the spatial feature maps and then fed back into the regressor for parameter rectification. Moreover, an auxiliary dense supervision is employed to enhance the learning of mesh-aligned features while spatial alignment attention is introduced to enable the awareness of the global contexts in our deep regressor. When extending PyMAF for full-body model recovery, an adaptive integration with the elbow-twist compensation strategy is proposed in PyMAF-X to produce natural wrist poses while maintaining the alignment performances of part-specific PyMAF. The efficacy of PyMAF and PyMAF-X is validated on indoor and in-the-wild datasets, where our approaches effectively improve the mesh-image alignment over the baseline and previous regression-based solutions.

Limitations and Future Work. In our experiments, we found that PyMAF-X fails to reconstruct interacting hands due to the separated regression of two hands. Meanwhile, when handling images with strong perspective distortions or with only upper-torso observations, common issues such as bent legs and erroneous limb poses remain unsolved in this work. We will leave these issues for future work to incorporate the merits of recent datasets [146], [147] and solutions such as SPEC [93], PIXIE [17], Hand4Whole [18], and IntagHand [10] into our framework.

Moreover, similar to existing methods [15], [16], [17], [18], the full-body alignment performance of PyMAF-X heavily relies on the pose and shape estimation of the body expert. Due to the lack of full-body mesh annotations, the estimated body shapes are typically inaccurate in challenging cases, resulting in erroneous bone lengths of arms and coarse alignment of hands. Combining PyMAF-X with SPIN [3], EFT [44], or NeuralAnnot [45] for the generation of more precise pseudo 3D ground-truth full-body mesh annotations on in-the-wild data would be interesting future work. Besides, the elbow-twist rotations are adjusted empirically in PyMAF-X based on the twist components of wrist poses. Learning the compensation angle α_{cp} via networks is also possible when large-scale full-body mesh annotations are available.

Acknowledgments

References

[1] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *CVPR*, 2018, pp. 7122–7131.

[2] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional mesh regression for single-image human shape reconstruction," in *CVPR*, 2019, pp. 4501–4510.

[3] N. Kolotouros, G. Pavlakos, M. J. Black, and K. Daniilidis, "Learning to reconstruct 3D human pose and shape via model-fitting in the loop," in *ICCV*, 2019, pp. 2252–2261.

[4] G. Moon and K. M. Lee, "I2L-MeshNet: Image-to-lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image," in *ECCV*. Springer, 2020, pp. 752–768.

[5] M. Kocabas, C.-H. P. Huang, O. Hilliges, and M. J. Black, "PARE: Part attention regressor for 3D human body estimation," in *ICCV*, 2021, pp. 11 127–11 137.

[6] H. Zhang, Y. Tian, X. Zhou, W. Ouyang, Y. Liu, L. Wang, and Z. Sun, "PyMAF: 3D human pose and shape regression with pyramidal mesh alignment feedback loop," in *ICCV*, 2021, pp. 11 446–11 456.

[7] X. Zhang, Q. Li, H. Mo, W. Zhang, and W. Zheng, "End-to-end hand mesh recovery from a monocular RGB image," in *ICCV*, 2019, pp. 2354–2364.

[8] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. Argus, and T. Brox, "FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images," in *ICCV*, 2019, pp. 813–822.

[9] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee, "Inter-Hand2.6M: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image," in *ECCV*. Springer, 2020, pp. 548–564.

[10] M. Li, L. An, H. Zhang, L. Wu, F. Chen, T. Yu, and Y. Liu, "Interacting attention graph for single image two-hand reconstruction," in *CVPR*, 2022, pp. 2761–2770.

[11] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3D face reconstruction and dense alignment with position map regression network," in *ECCV*, 2018, pp. 557–574.

[12] A. S. Jackson, A. Bulat, V. Argyriou, and G. Tzimiropoulos, "Large pose 3D face reconstruction from a single image via direct volumetric CNN regression," in *ICCV*, 2017, pp. 1031–1039.

[13] S. Sanyal, T. Bolkart, H. Feng, and M. J. Black, "Learning to regress 3D face shape and expression from an image without 3D supervision," in *CVPR*, 2019, pp. 7763–7772.

[14] Y. Feng, H. Feng, M. J. Black, and T. Bolkart, "Learning an animatable detailed 3D face model from in-the-wild images," *TOG*, vol. 40, no. 4, pp. 88:1–88:13, 2021.

[15] V. Choutas, G. Pavlakos, T. Bolkart, D. Tzionas, and M. J. Black, "Monocular expressive body regression through body-driven attention," in *ECCV*. Springer, 2020, pp. 20–40.

[16] Y. Rong, T. Shiratori, and H. Joo, "FrankMocap: A monocular 3D whole-body pose estimation system via regression and integration," in *ICCVW*, 2021.

[17] Y. Feng, V. Choutas, T. Bolkart, D. Tzionas, and M. J. Black, "Collaborative regression of expressive bodies using moderation," in *3DV*, 2021, pp. 792–804.

[18] G. Moon, H. Choi, and K. M. Lee, "Accurate 3D hand pose estimation for whole-body 3D human mesh estimation," in *CVPRW*, 2022.

[19] G. Pavlakos, L. Zhu, X. Zhou, and K. Daniilidis, "Learning to estimate 3D human pose and shape from a single color image," in *CVPR*, 2018, pp. 459–468.

[20] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "SMPL: A skinned multi-person linear model," *TOG*, vol. 34, no. 6, pp. 1–16, 2015.

[21] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3D hands, face, and body from a single image," in *CVPR*, 2019, pp. 10 975–10 985.

[22] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image," in *ECCV*. Springer, 2016, pp. 561–578.

[23] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. V. Gehler, "Unite the people: Closing the loop between 3D and 2D human representations," in *CVPR*, 2017, pp. 6050–6059.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *MICCAI*. Springer, 2015, pp. 234–241.

[25] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *ECCV*. Springer, 2016, pp. 483–499.

[26] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125.

[27] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," *CVPR*, 2019.

[28] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, "Neural body fitting: Unifying deep learning and model-based human pose and shape estimation," in *3DV*. IEEE, 2018, pp. 484–494.

[29] Y. Xu, S.-C. Zhu, and T. Tung, "DenseRaC: Joint 3D pose and shape estimation by dense render-and-compare," in *ICCV*, 2019, pp. 7760–7770.

[30] H. Zhang, J. Cao, G. Lu, W. Ouyang, and Z. Sun, "Learning 3D human shape and pose from dense body parts," *TPAMI*, 2020.

[31] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, "Human pose estimation with iterative error feedback," in *CVPR*, 2016, pp. 4733–4742.

[32] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *TOG*, vol. 36, no. 6, pp. 1–17, 2017.

[33] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *TOG*, vol. 36, no. 6, pp. 194–1, 2017.

[34] F. S. Grassia, "Practical parameterization of rotations using the exponential map," *JGT*, vol. 3, no. 3, pp. 29–48, 1998.

[35] Y. Tian, H. Zhang, Y. Liu, and L. Wang, "Recovering 3D human mesh from monocular images: A survey," *arXiv preprint arXiv:2203.01923*, 2022.

[36] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "SCAPE: shape completion and animation of people," *TOG*, vol. 24, pp. 408–416, 2005.

[37] L. Sigal, A. Balan, and M. J. Black, "Combined discriminative and generative articulated pose and non-rigid shape estimation," in *NeurIPS*, 2008, pp. 1337–1344.

[38] P. Guan, A. Weiss, A. O. Balan, and M. J. Black, "Estimating human shape and pose from a single image," in *ICCV*. IEEE, 2009, pp. 1381–1388.

[39] Y. Zhang, Z. Li, L. An, M. Li, T. Yu, and Y. Liu, "Lightweight multi-person total motion capture using sparse multi-view cameras," in *ICCV*, 2021, pp. 5560–5569.

[40] Y. Huang, F. Bogo, C. Lassner, A. Kanazawa, P. V. Gehler, J. Romero, I. Akhter, and M. J. Black, "Towards accurate markerless human shape and pose estimation over time," in *3DV*. IEEE, 2017, pp. 421–430.

[41] A. Zanfir, E. Marinoiu, and C. Sminchisescu, "Monocular 3D pose and shape estimation of multiple people in natural scenes - the importance of multiple scene constraints," in *CVPR*, 2018, pp. 2148–2157.

[42] R. A. Güler and I. Kokkinos, "HoloPose: Holistic 3D human reconstruction in-the-wild," in *CVPR*, 2019, pp. 10 884–10 894.

[43] J. Song, X. Chen, and O. Hilliges, "Human body model fitting by learned gradient descent," in *ECCV*. Springer, 2020, pp. 744–760.

[44] H. Joo, N. Neverova, and A. Vedaldi, "Exemplar fine-tuning for 3D human pose fitting towards in-the-wild 3D human pose estimation," in *3DV*, 2021, pp. 42–52.

[45] G. Moon, H. Choi, and K. M. Lee, "NeuralAnnot: Neural annotator for 3D human mesh training sets," in *CVPRW*, 2022, pp. 2299–2307.

[46] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *CVPR*, 2014, pp. 3686–3693.

[47] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation," in *BMVC*, 2010, pp. 12.1–12.11.

[48] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*. Springer, 2014, pp. 740–755.

[49] H. Choi, G. Moon, and K. M. Lee, "Pose2Mesh: Graph convolutional network for 3D human pose and mesh recovery from a 2D human pose," in *ECCV*, 2020.

[50] J. Li, C. Xu, Z. Chen, S. Bian, L. Yang, and C. Lu, "HybrIK: A hybrid analytical-neural inverse kinematics solution for 3D human pose and shape estimation," in *CVPR*, 2021, pp. 3383–3393.

[51] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *CVPR*, 2021, pp. 1954–1963.

[52] H.-Y. F. Tung, H.-W. Tung, E. Yumer, and K. Fragkiadaki, "Self-supervised learning of motion capture," *NeurIPS*, pp. 5236–5246, 2017.

[53] G. Pavlakos, N. Kolotouros, and K. Daniilidis, "TexturePose: Supervising human mesh estimation with texture consistency," in *ICCV*, 2019, pp. 803–812.

[54] Y. Rong, Z. Liu, C. Li, K. Cao, and C. C. Loy, "Delving deep into hybrid annotations for 3D human recovery in the wild," in *ICCV*, 2019, pp. 5340–5348.

[55] C. Doersch and A. Zisserman, "Sim2real transfer learning for 3D human pose estimation: motion to the rescue," *NeurIPS*, vol. 32, pp. 12 949–12 961, 2019.

[56] A. Zanfir, E. G. Bazavan, H. Xu, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "Weakly supervised 3D human pose and shape reconstruction with normalizing flows," in *ECCV*. Springer, 2020, pp. 465–481.

[57] J. N. Kundu, M. Rakesh, V. Jampani, R. M. Venkatesh, and R. V. Babu, "Appearance consensus driven self-supervised human mesh recovery," in *ECCV*. Springer, 2020, pp. 794–812.

[58] S. K. Dwivedi, N. Athanasiou, M. Kocabas, and M. J. Black, "Learning to regress bodies from images using differentiable semantic rendering," in *ICCV*, 2021, pp. 11 250–11 259.

[59] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid, "BodyNet: Volumetric inference of 3D human body shapes," in *ECCV*, 2018, pp. 20–36.

[60] M. Zanfir, A. Zanfir, E. G. Bazavan, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "THUNDR: Transformer-based 3D HUmaN reconstruction with markers," in *ICCV*, 2021.

[61] H. Choi, G. Moon, J. Park, and K. M. Lee, "Learning to estimate robust 3D human mesh from in-the-wild crowded scenes," in *CVPR*, 2022.

[62] N. Rueegg, C. Lassner, M. Black, and K. Schindler, "Chained representation cycling: Learning to estimate 3D human pose and shape by cycling between representations," in *AAAI*, vol. 34, 2020, pp. 5561–5569.

[63] A. Sengupta, I. Budvytis, and R. Cipolla, "Synthetic training for accurate 3D human pose and shape estimation in the wild," in *BMVC*, September 2020.

[64] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *CVPR*, 2017, pp. 1263–1272.

[65] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime multi-person 2D pose estimation using part affinity fields," *TPAMI*, vol. 43, no. 1, pp. 172–186, 2019.

[66] R. A. Güler, N. Neverova, and I. Kokkinos, "DensePose: Dense human pose estimation in the wild," in *CVPR*, 2018, pp. 7297–7306.

[67] S. Wu, S. Jin, W. Liu, L. Bai, C. Qian, D. Liu, and W. Ouyang, "Graph-based 3D multi-person pose estimation using multi-view images," in *ICCV*, 2021.

[68] Z. Zheng, T. Yu, Y. Wei, Q. Dai, and Y. Liu, "DeepHuman: 3D human reconstruction from a single image," in *ICCV*, 2019, pp. 7738–7748.

[69] K. Lin, L. Wang, and Z. Liu, "Mesh graphormer," in *ICCV*, 2021.

[70] P. Yao, Z. Fang, F. Wu, Y. Feng, and J. Li, "DenseBody: Directly regressing dense 3D human pose and shape from a single color image," *arXiv preprint arXiv:1903.10153*, 2019.

[71] T. Zhang, B. Huang, and Y. Wang, "Object-occluded human shape and pose estimation from a single color image," in *CVPR*, 2020, pp. 7376–7385.

[72] W. Zeng, W. Ouyang, P. Luo, W. Liu, and X. Wang, "3D human mesh regression with dense correspondence," in *CVPR*, 2020, pp. 7054–7063.

[73] Z. Wang, J. Yang, and C. Fowlkes, "The best of both worlds: Combining model-based and nonparametric approaches for 3D human body estimation," in *CVPRW*, 2022, pp. 2318–2327.

[74] M. Fieraru, M. Zanfir, E. Oneata, A.-I. Popa, V. Olaru, and C. Sminchisescu, "Three-dimensional reconstruction of human interactions," in *CVPR*, 2020, pp. 7212–7221.

[75] W. Jiang, N. Kolotouros, G. Pavlakos, X. Zhou, and K. Daniilidis, "Coherent reconstruction of multiple humans from a single image," in *CVPR*, 2020, pp. 5579–5588.

[76] J. Zhang, D. Yu, J. H. Liew, X. Nie, and J. Feng, "Body meshes as points," in *CVPR*, 2021, pp. 546–556.

[77] Y. Sun, Q. Bao, W. Liu, Y. Fu, M. J. Black, and T. Mei, "Monocular, one-stage, regression of multiple 3D people," in *ICCV*, 2021, pp. 11 179–11 188.

[78] Y. Sun, W. Liu, Q. Bao, Y. Fu, T. Mei, and M. J. Black, "Putting people in their place: Monocular regression of 3D people in depth," in *CVPR*, 2022, pp. 13 243–13 252.

[79] M. Fieraru, M. Zanfir, T. Szente, E. Bazavan, V. Olaru, and C. Sminchisescu, "REMIPS: Physically consistent 3D reconstruction of multiple interacting people under weak supervision," *NeurIPS*, vol. 34, pp. 19 385–19 397, 2021.

[80] M. Kocabas, N. Athanasiou, and M. J. Black, "VIBE: Video inference for human body pose and shape estimation," in *CVPR*, 2020, pp. 5253–5263.

[81] H. Choi, G. Moon, J. Y. Chang, and K. M. Lee, "Beyond static features for temporally consistent 3D human pose and shape from a video," in *CVPR*, 2021, pp. 1964–1973.

[82] Z. Wan, Z. Li, M. Tian, J. Liu, S. Yi, and H. Li, "Encoder-decoder with multi-level attention for 3D human shape and pose estimation," in *ICCV*, 2021, pp. 13 033–13 042.

[83] S. Zhang, Y. Zhang, F. Bogo, M. Pollefeys, and S. Tang, "Learning motion priors for 4D human body capture in 3D scenes," in *ICCV*, 2021, pp. 11 343–11 353.

[84] D. Rempe, T. Birdal, A. Hertzmann, J. Yang, S. Sridhar, and L. J. Guibas, "HuMoR: 3D human motion model for robust pose estimation," in *ICCV*, 2021.

[85] G. Pavlakos, J. Malik, and A. Kanazawa, "Human mesh recovery from multiple shots," in *CVPR*, 2022, pp. 1485–1495.

[86] R. Khirodkar, S. Tripathi, and K. Kitani, "Occluded human mesh recovery," in *CVPR*, 2022, pp. 1715–1725.

[87] Y. Yuan, U. Iqbal, P. Molchanov, K. Kitani, and J. Kautz, "GLAMR: Global occlusion-aware human mesh recovery with dynamic cameras," in *CVPR*, 2022, pp. 11 038–11 049.

[88] A. Sengupta, I. Budvytis, and R. Cipolla, "Hierarchical kinematic probability distributions for 3D human shape and pose estimation from images in the wild," in *ICCV*, 2021, pp. 11 219–11 229.

[89] V. Choutas, L. Müller, C.-H. P. Huang, S. Tang, D. Tzionas, and M. J. Black, "Accurate 3D body shape regression using metric and semantic attributes," in *CVPR*, 2022, pp. 2718–2728.

[90] A. Sengupta, I. Budvytis, and R. Cipolla, "Probabilistic 3D human shape and pose estimation from multiple unconstrained images in the wild," in *CVPR*, 2021, pp. 16 094–16 104.

[91] N. Kolotouros, G. Pavlakos, D. Jayaraman, and K. Daniilidis, "Probabilistic modeling for human mesh recovery," in *ICCV*, 2021, pp. 11 605–11 614.

[92] I. Kissos, L. Fritz, M. Goldman, O. Meir, E. Oks, and M. Kliger, "Beyond weak perspective for monocular 3D human pose estimation," in *ECCV*. Springer, 2020, pp. 541–554.

[93] M. Kocabas, C.-H. P. Huang, J. Tesch, L. Muller, O. Hilliges, and M. J. Black, "SPEC: Seeing people in the wild with an estimated camera," in *ICCV*, 2021, pp. 11 035–11 045.

[94] S. Guan, J. Xu, Y. Wang, B. Ni, and X. Yang, "Bilevel online adaptation for out-of-domain human mesh reconstruction," in *CVPR*, 2021, pp. 10 472–10 481.

[95] Y. Rong, Z. Liu, and C. C. Loy, "Chasing the tail in monocular 3D human reconstruction with prototype memory," *TIP*, vol. 31, pp. 2907–2919, 2022.

[96] S. Saito, Z. Huang, R. Natsume, S. Morishima, A. Kanazawa, and H. Li, "PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization," in *ICCV*, 2019, pp. 2304–2314.

[97] Z. Zheng, T. Yu, Y. Liu, and Q. Dai, "PaMIR: Parametric model-conditioned implicit representation for image-based human reconstruction," *TPAMI*, 2021.

[98] Y. Zheng, R. Shao, Y. Zhang, T. Yu, Z. Zheng, Q. Dai, and Y. Liu, "DeepMultiCap: Performance capture of multiple characters using sparse multiview cameras," in *ICCV*, 2021, pp. 6239–6249.

[99] Y. Xiu, J. Yang, D. Tzionas, and M. J. Black, "ICON: Implicit clothed humans obtained from normals," in *CVPR*, 2022, pp. 13 296–13 306.

[100] S. Baek, K. I. Kim, and T.-K. Kim, "Pushing the envelope for RGB-based dense 3D hand pose estimation via neural rendering," in *CVPR*, 2019, pp. 1067–1076.

[101] A. Boukhayma, R. d. Bem, and P. H. Torr, "3D hand shape and pose from images in the wild," in *CVPR*, 2019, pp. 10 843–10 852.

[102] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid, "Learning joint reconstruction of hands and manipulated objects," in *CVPR*, 2019, pp. 11 807–11 816.

[103] D. Kulon, H. Wang, R. A. Güler, M. M. Bronstein, and S. Zafeiriou, "Single image 3D hand reconstruction with mesh convolutions," in *BMVC*, 2019.

[104] L. Ge, Z. Ren, Y. Li, Z. Xue, Y. Wang, J. Cai, and J. Yuan, "3D hand shape and pose estimation from a single RGB image," in *CVPR*, 2019, pp. 10 833–10 842.

[105] D. Kulon, R. A. Güler, I. Kokkinos, M. M. Bronstein, and S. Zafeiriou, "Weakly-supervised mesh-convolutional hand reconstruction in the wild," in *CVPR*, 2020, pp. 4989–4999.

[106] B. Zhang, Y. Wang, X. Deng, Y. Zhang, P. Tan, C. Ma, and H. Wang, "Interacting two-hand 3D pose and shape reconstruction from single color image," in *ICCV*, 2021, pp. 11 354–11 363.

[107] Y. Rong, J. Wang, Z. Liu, and C. C. Loy, "Monocular 3D reconstruction of interacting hands via collision-aware factorized refinements," in *3DV*, 2021, pp. 432–441.

[108] A. Tewari, M. Zollhöfer, P. Garrido, F. Bernard, H. Kim, P. Pérez, and C. Theobalt, "Self-supervised multi-level face model learning for monocular reconstruction at over 250 Hz," in *CVPR*, 2018, pp. 2549–2559.

[109] L. Wang, Z. Chen, T. Yu, C. Ma, L. Li, and Y. Liu, "FaceVerse: a fine-grained and detail-controllable 3D face morphable model from a hybrid dataset," in *CVPR*, 2022, pp. 20 333–20 342.

[110] H. Joo, T. Simon, and Y. Sheikh, "Total capture: A 3D deformation model for tracking faces, hands, and bodies," in *CVPR*, 2018, pp. 8320–8329.

[111] H. Xu, E. G. Bazavan, A. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "GHUM & GHUML: Generative 3D human shape and articulated pose models," in *CVPR*, 2020, pp. 6184–6193.

[112] D. Xiang, H. Joo, and Y. Sheikh, "Monocular total capture: Posing face, body, and hands in the wild," in *CVPR*, 2019, pp. 10 965–10 974.

[113] K. Li, Y. Mao, Y. Liu, R. Shao, and Y. Liu, "Full-body motion capture for multiple closely interacting persons," *GM*, vol. 110, p. 101072, 2020.

[114] Y. Sun, T. Huang, Q. Bao, W. Liu, W. Gao, and Y. Fu, "Learning monocular mesh recovery of multiple body parts via synthesis," in *ICASSP*, 2022, pp. 2669–2673.

[115] Y. Zhou, M. Habermann, I. Habibie, A. Tewari, C. Theobalt, and F. Xu, "Monocular real-time full body capture with inter-part correlations," in *CVPR*, 2021, pp. 4811–4822.

[116] A. Zanfir, E. G. Bazavan, M. Zanfir, W. T. Freeman, R. Sukthankar, and C. Sminchisescu, "Neural descent for visual 3D human pose and shape," in *CVPR*, 2021, pp. 14 484–14 493.

[117] I. Grishchenko, V. Bazarevsky, A. Zanfir, E. G. Bazavan, M. Zanfir, R. Yee, K. Raveendran, M. Zhdanovich, M. Grundmann, and C. Sminchisescu, "Blazepose GHUM holistic: Real-time 3D human landmarks and pose estimation," in *CVPRW*, 2022.

[118] C. Nakatsuka, J. Xu, and K. Tasaka, "Learning joint twist rotation for 3D human pose estimation from a single image." in *VISAPP*, 2021, pp. 379–386.

[119] H. Zhu, X. Zuo, S. Wang, X. Cao, and R. Yang, "Detailed human shape estimation from a single image by hierarchical mesh deformation," in *CVPR*, 2019, pp. 4491–4500.

[120] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *CVPR*, 2013, pp. 532–539.

[121] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou, "Mnemonic descent method: A recurrent process applied for end-to-end face alignment," in *CVPR*, 2016, pp. 4177–4187.

[122] P. Chandran, D. Bradley, M. Gross, and T. Beeler, "Attention-driven cropping for very high resolution facial landmark detection," in *CVPR*, 2020, pp. 5861–5870.

[123] M. Oberweger, P. Wohlhart, and V. Lepetit, "Generalized feedback loop for joint hand-object pose estimation," *TPAMI*, vol. 42, no. 8, pp. 1898–1912, 2019.

[124] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2Mesh: Generating 3D mesh models from single RGB images," in *ECCV*, 2018, pp. 52–67.

[125] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[126] X. Zhu, X. Liu, Z. Lei, and S. Z. Li, "Face alignment in full pose range: A 3D total solution," *TPAMI*, vol. 41, no. 1, pp. 78–92, 2017.

[127] B. Huyghe, "Design and implementation of a mobile sensor system for human posture tracking," Ph.D. dissertation, Ghent University, 2011.

[128] P. Dobrowolski, "Swing-twist decomposition in clifford algebra," *arXiv preprint arXiv:1506.05481*, 2015.

[129] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[130] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *CVPR*, 2019, pp. 5745–5753.

[131] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2014.

[132] P. Patel, C.-H. P. Huang, J. Tesch, D. T. Hoffmann, S. Tripathi, and M. J. Black, "AGORA: Avatars in geography optimized for regression analysis," in *CVPR*, 2021, pp. 13468–13478.

[133] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3D human pose in the wild using IMUs and a moving camera," in *ECCV*, 2018, pp. 601–617.

[134] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *TPAMI*, vol. 36, no. 7, pp. 1325–1339, 2014.

[135] A. Kanazawa, J. Y. Zhang, P. Felsen, and J. Malik, "Learning 3D human dynamics from video," in *CVPR*, 2019, pp. 5614–5623.

[136] A. Arnab, C. Doersch, and A. Zisserman, "Exploiting temporal context for 3D human pose estimation in the wild," in *CVPR*, 2019, pp. 3395–3404.

[137] Y. Sun, Y. Ye, W. Liu, W. Gao, Y. Fu, and T. Mei, "Human mesh recovery from monocular images via a skeleton-disentangled representation," in *ICCV*, 2019, pp. 5349–5358.

[138] Z. Luo, S. A. Golestaneh, and K. M. Kitani, "3D human motion estimation via motion compression and refinement," in *ACCV*, 2020.

[139] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3D human pose estimation in the wild using improved CNN supervision," in *3DV*, 2017, pp. 506–516.

[140] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *CVPR*, 2011, pp. 1465–1472.

[141] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu, C. Qian, W. Ouyang, and P. Luo, "Whole-body human pose estimation in the wild," in *ECCV*, 2020, pp. 196–214.

[142] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," in *FG*, 2018, pp. 67–74.

[143] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks)," in *ICCV*, 2017, pp. 1021–1030.

[144] Z.-H. Feng, P. Huber, J. Kittler, P. Hancock, X.-J. Wu, Q. Zhao, P. Koppen, and M. Rätsch, "Evaluation of dense 3D reconstruction from 2D face images in the wild," in *FG*, 2018, pp. 780–786.

[145] S. Kreiss, L. Bertoni, and A. Alahi, "OpenPifPaf: Composite fields for semantic keypoint detection and spatio-temporal association," *TPAMI*, 2021.

[146] L. Müller, A. A. Osman, S. Tang, C.-H. P. Huang, and M. J. Black, "On self-contact and human pose," in *CVPR*, 2021, pp. 9990–9999.

[147] C.-H. P. Huang, H. Yi, M. Höschle, M. Safroshkin, T. Alexiadis, S. Polikovsky, D. Scharstein, and M. J. Black, "Capturing and inferring dense full-body human-scene contact," in *CVPR*, 2022, pp. 13274–13285.

[148] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, "Accelerating 3D deep learning with PyTorch3D," *arXiv preprint arXiv:2007.08501*, 2020.

[149] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *CVPR*, 2019, pp. 4401–4410.

[150] M. Loper, N. Mahmood, and M. J. Black, "MoSh: Motion and shape capture from sparse markers," *TOG*, vol. 33, no. 6, pp. 1–13, 2014.
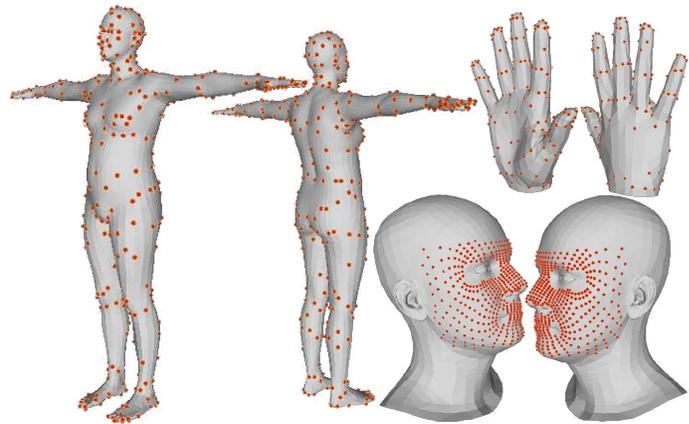
Fig. 10. Visualization of the selected vertices on the body, hand, and face models for mesh-aligned feature extraction.
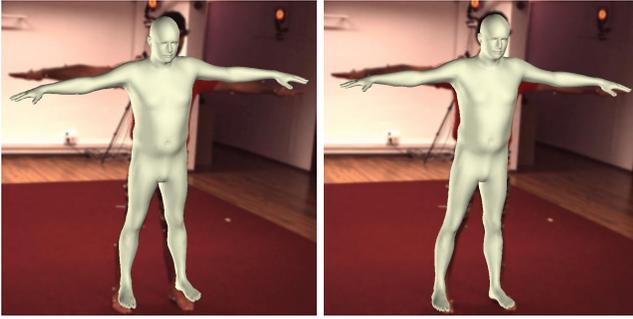

Fig. 11. Visualization of the rendered dense correspondences in resolutions of 56×56 for the body, hand, and face meshes. From left to right: the IUV [66] map for a body model, the PNCC [126] maps for hand and face models.

APPENDIX A
MORE IMPLEMENTATION DETAILS

Attention Module. We follow Mesh Graphormer [69] to implement the attention module for the grid and mesh-aligned feature fusion. We simply use a single attention block for each iteration because we found that a single block brought enough performance grains while using more attention blocks merely increased memory consumption.

Mesh Sampling. For body and hand meshes, we simply use the down-sampling matrix provided in GraphCMR [2] and Mesh Graphormer [69] to reduce the vertex number for mesh-aligned feature extraction. For face meshes, we manually select the vertices in the front face region considering that the expression information concentrates on the front face. For body-only PyMAF, the dimension of the mesh-aligned features is reduced to match the feature dimension in HMR [1] for more fair comparisons. For hand- and face-only PyMAF, we do not strictly abide by this rule but simply replace SMPL with MANO/FLAME during the implementation. Fig. 10 visualizes the selected vertices on the body, hand, and face meshes.

Auxiliary Dense Supervision. In the auxiliary dense prediction task, the dense correspondence and part segmen-

(a) PA-MPJPE: 26.9, MPJPE: 74.3   (b) PA-MPJPE: 27.7, MPJPE: 43.4

Fig. 12. Examples of two reconstruction results. (a) A reconstruction result with a lower PA-MPJPE value but worse mesh-image alignment. (b) A reconstruction result with a higher PA-MPJPE value but better mesh-image alignment.

tation (for ablation experiments) used for supervision are rendered based on the pseudo ground truth meshes. Examples of the rendered dense correspondence are visualized in Fig. 11. We choose to use the rendered dense correspondence mainly based on the following reasons:

i) The dense correspondence can be regarded as a more fine-grained part segmentation. In our method, we use the IUV representation as dense correspondence for the body expert, while using the Projected Normalized Coordinate Code (PNCC) [126] for hand and face experts. Dense correspondences are more general since PNCC does not need to split the mesh into manually defined parts.

ii) The body part definition may vary from different datasets while the rendered one is consistent. Moreover, the rendering process can be done efficiently in batch using the tools like PyTorch3D [148]. The costs of rendering part segmentation and dense correspondences are almost the same for the PyTorch3D renderer.

iii) There is only a limited number of datasets providing the annotated part segmentation and dense correspondence, while the pseudo ground truth meshes are typically available for common datasets thanks to previous work such as SPIN [3] and EFT [44].

Hand Visibility. We simply use three FC layers to predict hand visibility based on the hand-only mesh-aligned features. The pseudo visibility confidence used for supervision is calculated based on the proportion of the visible hand keypoints annotated in the whole-body COCO [141] dataset. For example, assume that there are 15 visible keypoints and the total number of hand keypoints is 21, then the pseudo visibility confidence of this hand is about 0.714 (15/21).

APPENDIX B
ABOUT METRICS

Though the PA-PVE and PA-MPJPE are widely adopted in the 3D pose estimation task, these two metrics can not fully reveal the mesh-image alignment performance since they are calculated after rigid alignment. As depicted in Fig. 12, a reconstruction result with a lower PA-MPJPE value can have a higher MPJPE value and worse alignment between the reprojected mesh and image.

APPENDIX C
ABOUT DATASETS

Following the practices of previous work [1], [3], [5], we train our body model regression network on several datasets with 3D or 2D annotations, including Human3.6M [134], MPI-INF-3DHP [139], LSP [47], MPII [46], COCO [48]. For hand-only and full-body model regression, FreiHAND [8], InterHand2.6M [9], FFHQ [149], and COCO-WholeBody [141] are also used for training. Here, we provide more descriptions of the datasets to supplement the main manuscript.

Human3.6M [134] is commonly used as the benchmark dataset for 3D human pose estimation, consisting of 3.6 million video frames captured in the controlled environment. The ground truth SMPL parameters in Human3.6M are generated by applying MoSH [150] to the sparse 3D MoCap marker data, as done in Kanazawa et al. [1]. The original videos are down-sampled from 50fps to 10fps, resulting in 312,188 frames for training. Following the common protocols [1], [19], [64], our experiments use five subjects (S1, S5, S6, S7, S8) for training and two subjects (S9, S11) for evaluation. The original videos are also down-sampled from 50 fps to 10 fps to remove redundant frames, resulting in 312,188 frames for training and 26,859 frames for evaluation.

3DPW [133] is captured in challenging outdoor scenes with IMU-equipped actors under various activities. This dataset provides accurate shape and pose ground truth annotations. Following the protocol of previous work [3], [135], we do not use its data for training by default unless specified in the table.

MPI-INF-3DHP [139] is a 3D human pose dataset covering more actor subjects and poses than Human3.6M. The images of this dataset were collected under both indoor and outdoor scenes, and the 3D annotations were captured by a multi-camera marker-less MoCap system. Hence, there is some noise in the 3D ground truth annotations. The training set includes 8 subjects and there are 96,507 frames down-sampled from videos used for training.

LSP [47] and **LSP-Extended** [140] are 2D human pose benchmark datasets, containing person images with challenging poses. There are 14 visible 2D keypoint locations annotated for each image and 10,428 samples used for training.

MPII [46] is a standard benchmark for 2D human pose estimation. There are 25,000 images collected from YouTube videos covering a wide range of activities. We discard those images without complete keypoint annotations, producing 14,667 samples for training.

COCO [48] and **COCO-WholeBody** [141] contain a large scale of person images labeled with 17 body keypoints, 42 hand keypoints, and 68 face keypoints. We use COCO to train body-only PyMAF and leverage the hand keypoints in COCO-WholeBody during the training of hand- and face-only PyMAF and PyMAF-X. Since this dataset does not contain ground-truth meshes, we conduct a quantitative evaluation on the 2D keypoint localization task using its validation set, which consists of 50,197 samples.

EHF [21] contains 100 testing images of one subject captured in lab environments. For each image, the corresponding 3D scans and ground-truth SMPL-X [21] meshes

are provided. EHF is used for testing only and is commonly adopted as a full-body evaluation benchmark dataset in literature [15], [17], [18].

AGORA [132] is a synthetic dataset with accurate SMPL-X models fitted to 3D scans. Since the ground-truth labels of its test set are not publicly available, the evaluation is performed on the official platform[2]. For evaluation on AGORA, we use the training set of AGORA to finetune our model.

FreiHAND [8] contains 130,240 samples for training and 3960 samples images for evaluation. For each sample in the training set, the MANO [32] parameters recovered from multi-view images are provided. We use this dataset for the training and evaluation of the hand expert.

InterHand2.6M [9] is a large-scale real-captured hand dataset, providing accurate MANO parameters of interacting hands. We crop single-hand images from this dataset for the training of the hand expert.

VGGFace2 [142] is a large-scale face dataset. The images of this dataset are downloaded from Google and have large variations in pose, age, and ethnicity. It contains about 3 million images from training. We run the method of FAN [143] and DECA [14] on its training set to generate the pseudo ground truth facial landmarks and FLAME [33] models for the training of the face expert.

Stirling3D [144] provides facial images with the ground-truth 3D scans. The test set contains 2,000 facial images in neutral expressions, including 1,344 low-quality (LQ) images and 656 high-quality (HQ) images. We follow previous work [14], [15] to use it for evaluation only.

NoW [13] contains the facial images captured with an iPhone X, and a separate 3D scan for each subject. Its test set contains 1,702 images for evaluation. Since the ground-truth scans of the test set are not publicly available, the evaluation is performed by following the instructions on the official website[3]. We follow previous work [14], [15] to use this dataset for evaluation only.

APPENDIX D
MORE QUALITATIVE RESULTS

We provide more qualitative results of our method in this section. In Fig 13, we visualize the estimated meshes after each iteration, where it can be seen that PyMAF can correct the drift of body parts progressively and result in better-aligned human models. In Fig. 14, the body mesh recovery results of different methods on COCO are depicted for qualitative comparisons, where PyMAF convincingly performs better than competitors and our baseline by producing better-aligned and natural results. In Fig. 15, we provide more full-body model reconstruction results on the COCO validation set, where PyMAF-X can produce well-aligned full-body model under challenging cases. In Fig.16, we further visualize the reconstructed full-body models from different viewpoints.

Cases under Occlusions. As pointed out in the main paper, the adaptive integration is not applicable when the hand part is invisible. To handle this, the visibility status of
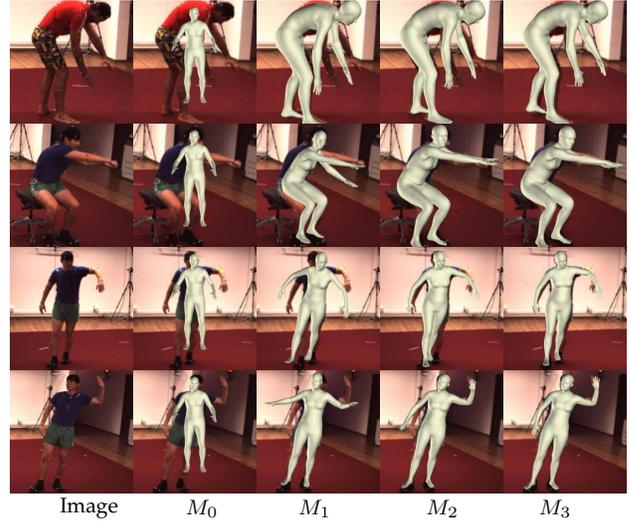
Fig. 13. Visualization of reconstruction results across different iterations in the feedback loop.

Fig. 14. Qualitative comparison of reconstruction results on the COCO validation set.

hands is also predicted by the hand expert in PyMAF-X. In cases of invisible hands, the full-body model adopts the default hand poses and the wrist poses estimated by the body expert. Fig. 17 shows the example results of PyMAF-X when the body or hands are occluded. We can see that PyMAF-X produces reasonable full-body meshes under these cases.

Failure Cases. Due to the rotational pose representation of the kinematic model, the full-body alignment of PyMAF-X heavily relies on the accuracy of body pose estimation. Moreover, the misalignment may also occur when the body shape is inaccurate since it affects the body bone length. Besides, it is still challenging for PyMAF-X to handle challenging hand poses or interacting hands. Fig. 18 visualizes some erroneous results of our approach, where PyMAF-X produces misaligned results due to the issues mentioned above.

Fig. 15. More examples of the full-body mesh recovery results of PyMAF-X on the COCO validation set. Best viewed zoomed-in on a color screen.

Fig. 16. PyMAF-X results visualized from different viewpoints. For each example, from left to right: the input image, the overlay result, and the results with rotations around the vertical axis. Best viewed zoomed-in on a color screen.

Fig. 17. Example results of PyMAF-X when the body or hands are occluded. Samples come from the COCO validation set. Best viewed zoomed-in on a color screen.

(a) Inaccurate bone length (body shape)

(b) Inaccurate body pose

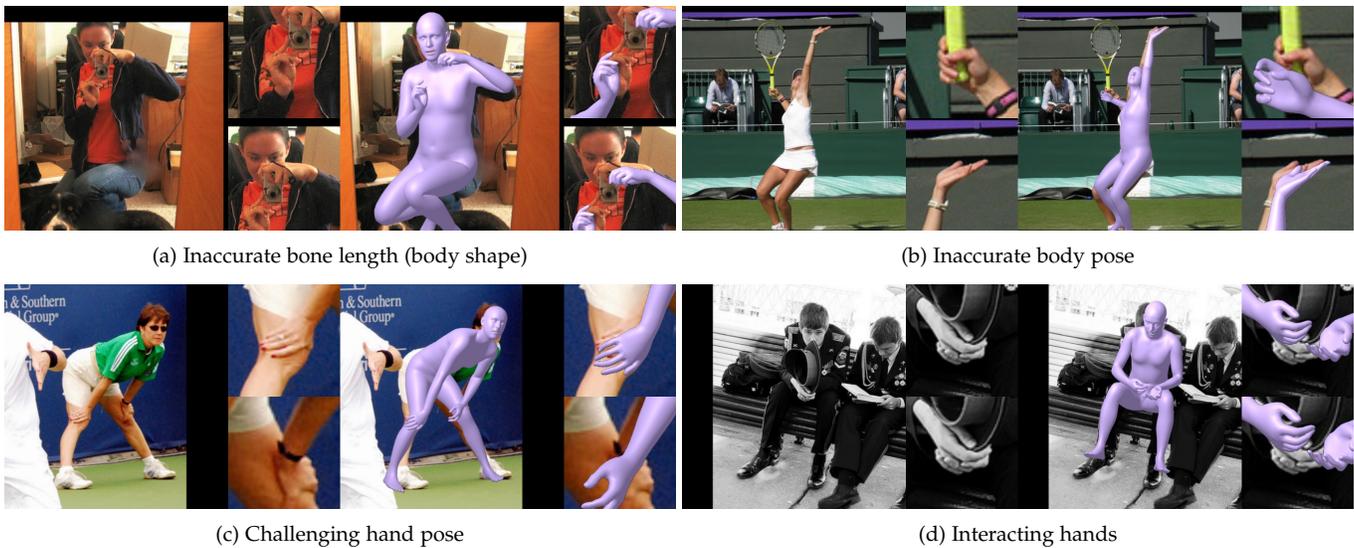(c) Challenging hand pose

(d) Interacting hands

Fig. 18. Misaligned reconstructions of our approach. Misalignment comes from (a) inaccurate bone length (body shape), (b) inaccurate body pose, (c)(d) inaccurate hand poses under challenging hand poses, occlusions, and interactions. Samples come from the COCO validation set.