Three Recipes for Better 3D Pseudo-GTs of 3D Human Mesh Estimation in the Wild

Gyeongsik Moon[1] Hongsuk Choi[2] Sanghyuk Chun[3] Jiyoung Lee[3] Sangdoo Yun[3]

[1] Meta Reality Labs [2] Samsung AI Center - New York [3] NAVER AI Lab

mks0601@gmail.com redstonepo@gmail.com {sanghyuk.c, lee.j, sangdoo.yun}@navercorp.com

Abstract

Recovering 3D human mesh in the wild is greatly challenging as in-the-wild (ITW) datasets provide only 2D pose ground truths (GTs). Recently, 3D pseudo-GTs have been widely used to train 3D human mesh estimation networks as the 3D pseudo-GTs enable 3D mesh supervision when training the networks on ITW datasets. However, despite the great potential of the 3D pseudo-GTs, there has been no extensive analysis that investigates which factors are important to make more beneficial 3D pseudo-GTs. In this paper, we provide three recipes to obtain highly beneficial 3D pseudo-GTs of ITW datasets. The main challenge is that only 2D-based weak supervision is allowed when obtaining the 3D pseudo-GTs. Each of our three recipes addresses the challenge in each aspect: depth ambiguity, sub-optimality of weak supervision, and implausible articulation. Experimental results show that simply re-training state-of-the-art networks with our new 3D pseudo-GTs elevates their performance to the next level without bells and whistles. The 3D pseudo-GT is publicly available[1].

1. Introduction

3D human mesh estimation aims to localize 3D human mesh vertices in the 3D space. The major challenge is the lack of 3D ground truths (GTs) of in-the-wild (ITW) datasets [1, 16, 28]. Images of ITW datasets are captured with a single camera without special equipment, such as inertial measurement units (IMUs) and multiple calibrated cameras, as ITW images are taken in our daily life. As such special equipment is necessary to obtain 3D mesh data, only sparse 2D GT poses (*i.e.*, 2D GT coordinates of about twenty joints) are available in ITW datasets without 3D dense mesh GTs that have thousands of vertices.

The main training strategy for the 3D human mesh estimation in the wild is a mixed-batch training [6–8, 21, 22, 26, 27, 33, 35, 40], which takes half samples of a mini-batch from motion capture (MoCap) datasets [14, 17, 31, 37, 45] and rest samples from ITW datasets. MoCap datasets are captured from a controlled environment, such as a lab or studio, and they provide 3D pose and mesh GTs by utilizing special equipment, such as multiple calibrated cameras. During the mixed-batch training, samples from MoCap datasets are supervised with 3D GT meshes, and those from ITW datasets are supervised with 3D pseudo-GT meshes.

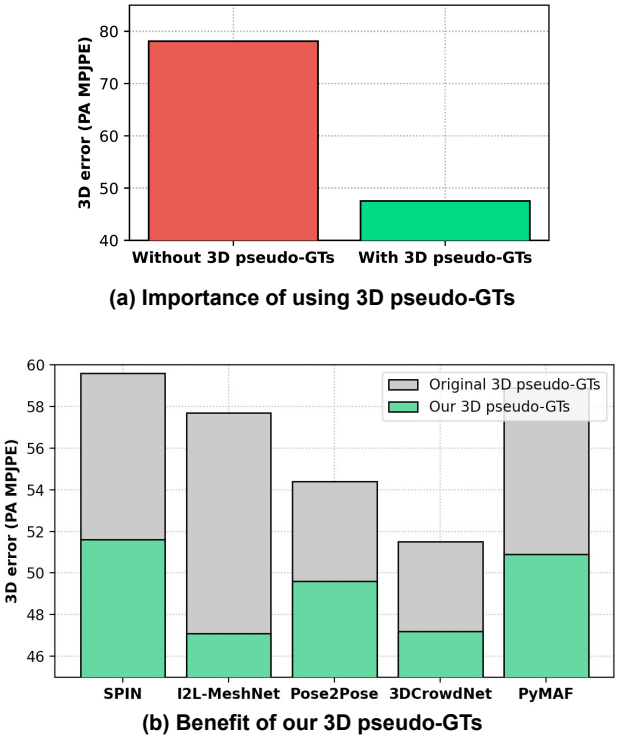

Figure 1. (a) 3D error (PA MPJPE) comparison on 3DPW [42] between Pose2Pose [33] trained without and with 3D pseudo-GTs. (b) 3D error (PA MPJPE) comparison on 3DPW [42] between networks trained with their and our 3D pseudo-GTs. The numbers are from Table 3.

[1] https://github.com/mks0601/NeuralAnnot_RELEASE

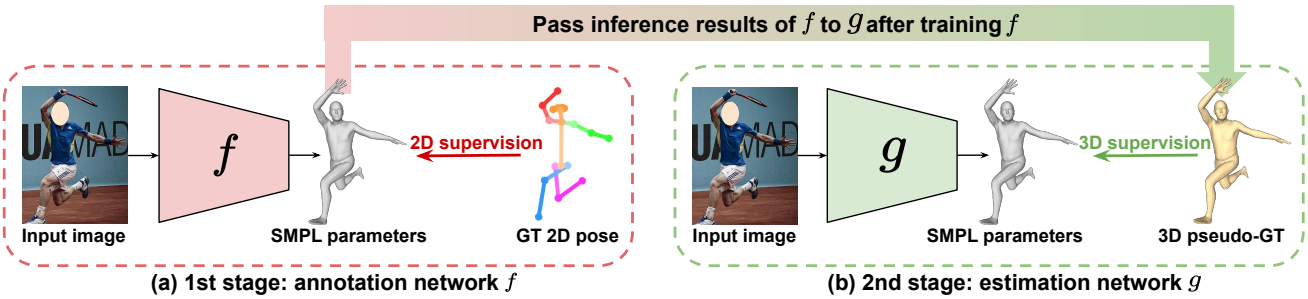

Figure 2. The overall pipeline of the proposed framework. (a): In the first stage, the annotation network f outputs SMPL parameters, *weakly supervised with 2D GT pose*. (b): After finishing training the annotation network f, the inference results of f on seen training sets become 3D pseudo-GTs. In the second stage, the estimation network g is *fully supervised with the 3D pseudo-GTs*. For simplicity, we do not depict 3D supervisions of MoCap datasets during the mixed-batch training of f and g.

The contribution of MoCap datasets is providing 3D supervision with their accurate 3D GTs, which do not exist in ITW datasets. However, using MoCap datasets is not sufficient for the best performance in the wild. This is because they are collected from the controlled environment; therefore, their image appearances, such as illumination and backgrounds, are highly limited and far from those of ITW datasets [7, 8, 36]. To cope with such limitation, 3D pseudo-GTs of ITW datasets have been widely used to provide 3D supervision to ITW samples. Although 3D pseudo-GTs contain errors in nature, they provide 3D supervision to ITW samples, which can complement 2D-based weak supervision from 2D GT poses of ITW datasets.

Fig. 1 (a) shows that the 3D pseudo-GTs of ITW datasets boost the performance a lot compared to a counterpart that does not utilize the 3D pseudo-GTs. The figure shows that 3D pseudo-GTs are greatly important for high performance and justifies the **two-stage training pipeline** (Fig. 2), of which the **first stage** is acquiring 3D pseudo-GTs, and the **second stage** is training a 3D human mesh estimation network [6–8, 21, 22, 26, 27, 33, 35, 40] with the 3D pseudo-GTs. In the first stage, the 3D pseudo-GTs are acquired using either the iterative fitting framework [3, 38] or *external* annotation network [18, 23, 34]. We denote the annotation network of the first stage by f and the estimation network of the second stage by g.

Annotation networks f [3, 18, 23, 34, 38] are weakly supervised with 2D GT poses to obtain 3D pseudo-GTs of ITW datasets. The weak supervision of ITW samples is enabled by SMPL body model [29], which produces 3D human mesh from pose and shape parameters in a differentiable way. After extracting 3D joint coordinates from the 3D mesh and projecting them to the 2D space, the 2D-based weak supervision minimizes the distance between the projected 2D joint coordinates and 2D GT pose. In this way, the 2D GT pose weakly supervises SMPL parameters, which can make all vertices of the 3D mesh fit to the 2D GT pose. In this paper, we define 3D pseudo-GTs as SMPL parameters.

Unfortunately, although many recent 3D human mesh estimation methods train their networks g [6–8, 21, 22, 26, 27, 33, 35, 40] with 3D pseudo-GTs of ITW datasets for their performances, there has been no extensive analysis that investigates which factors are important to obtain beneficial 3D pseudo-GTs. **In this paper, we provide three recipes for highly beneficial 3D pseudo-GTs of ITW datasets.** The main challenge is that *only 2D-based weak supervision is allowed* without 3D evidence in ITW datasets when obtaining the 3D pseudo-GTs. The absence of the 3D evidence when training the annotation networks f (*i.e.*, the first stage in Fig. 2) causes severe ambiguities, while the estimation networks g (*i.e.*, the second stage in Fig. 2) suffer less from them as the 3D pseudo-GTs from the first stage serve 3D evidence.

We address the challenge of obtaining beneficial 3D pseudo-GTs (*i.e.*, the first stage in Fig. 2) in three aspects: *depth ambiguity, sub-optimality of weak supervision, and implausible articulation.* First, multiple 3D data (*e.g.*, SMPL parameters) corresponds to the same 2D evidence, which incurs depth ambiguity. Second, weak supervision signals make networks converge to sub-optimal points compared to full supervision. Finally, 3D human meshes with anatomically implausible articulations can correspond to the 2D GT pose. All the previous iterative fitting frameworks [3, 38] and annotation networks f [18, 23, 34] suffer from the problems as they rely on the 2D-based weak supervision when obtaining 3D pseudo-GTs; however, they have not carefully considered the problems. Fig. 1 (b) shows that without bells and whistles, simply re-training state-of-the-art estimation networks g with our new 3D pseudo-GTs elevate their performance to the next level on ITW benchmarks [42]. Fig. 3 shows that the performance of estimation network g improves with each recipe applied. We will publicly open our 3D pseudo-GTs, which can benefit the community and following works.

Annotation networks f	Train f on 3DPW	Initialization of f	Use VPoser and L2 reg. in f
SPIN [23]	✗	ImageNet classification [12]	✗
EFT [18]	✗	3D pose network [23]	✗
NeuralAnnot [34]	✗	ImageNet classification [12]	✓
Ours	✓	2D pose network [44]	✓

Table 1. Comparison of previous annotation networks and ours.

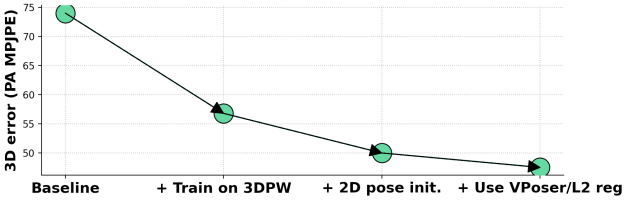

Figure 3. 3D errors (PA MPJPE) of estimation network g, trained with different 3D pseudo-GTs, on 3DPW [42]. The lower the better. Each pseudo-GT is designed for mitigating depth ambiguity, sub-optimality of weak supervision, and implausible articulation. Our three recipes significantly improve 3D errors of g.

Figure 4. Comparisons of images from MoCap dataset [14], 3DPW [42], and ITW dataset [28].

2. 3D pseudo-GTs of ITW datasets

2.1. Overall pipeline

Fig. 2 shows the overall pipeline of the proposed framework. Our entire system consists of two networks: annotation network f and estimation network g, where both networks are trained with the mixed-batch training strategy. The annotation network f is trained with 2D and 3D GTs of ITW and MoCap datasets, respectively. Please note that the mixed-batch training of the annotation network f is different from that of estimation network g in that only 2D supervision, without 3D supervision, is available for ITW samples. The testing results of f on seen training images of ITW datasets become 3D pseudo-GTs. The 3D pseudo-GTs are used to train the estimation network g. As developing a new network architecture is not our focus, we design the annotation network f to have the network architecture of Pose2Pose [33], a state-of-the-art SMPL parameter regression network. For the details of Pose2Pose, please refer to the supplementary material. We use various state-of-the-art 3D human mesh estimation networks [8, 22, 23, 26, 33, 35] for the estimation network g and show generalizability of our 3D pseudo-GTs to them in the experimental section.

2.2. Three recipes for 3D pseudo-GTs

The major challenge to obtaining beneficial 3D pseudo-GTs of ITW datasets is that only weak supervision targets (*i.e.*, 2D GT poses) are available without 3D evidence. The absence of the 3D evidence when training the annotation networks f (*i.e.*, the first stage in Fig. 2) causes severe ambiguities, while the estimation networks g (*i.e.*, the second stage in Fig. 2) suffers less from the ambiguities as the 3D pseudo-GTs from the first stage serve 3D evidence. We design our recipes to address the challenge of obtaining more beneficial 3D pseudo-GTs in three aspects: *depth ambiguity, sub-optimality of weak supervision, and implausible articulation*. Fig. 3 shows how the 3D error of the estimation network g changes when the 3D pseudo-GTs of ITW datasets are obtained following our recipes. Our three recipes are summarized below.

1. To resolve the depth ambiguity, even if the scales of datasets are small, collect ITW datasets with 3D GTs (e.g., 3DPW [42]) and train the annotation network f on them. The 2D-based weak supervision causes depth ambiguity as there can be an infinite number of 3D data (*e.g.*, SMPL parameters) that correspond to the same 2D evidence. Previous annotation networks f [18, 23] alleviated the depth ambiguity by using MoCap datasets during the mixed-batch training. As MoCap datasets provide 3D GT meshes, their networks learn an image-to-3D mesh function from MoCap datasets, and the learned function is shared with the ITW case in the same network. However, it is not sufficient as MoCap images have largely different image appearances, such as backgrounds, illuminations, and colors, compared to those of ITW images. The reason for such a large appearance gap is that MoCap datasets are captured from a restricted environment, such as a studio or lab, while ITW datasets are captured from anywhere in our daily life. Due to such a large appearance gap, knowledge learned from MoCap samples might not sufficiently be transferred to the ITW case.

To bridge MoCap and ITW datasets, even if the scales of datasets are small, we propose to collect ITW datasets with 3D GTs and train the annotation network f on them. One example of such a small-scale ITW dataset with 3D GTs is 3DPW [42]. 3DPW is captured from the outdoor environment with moving cameras, and its image appearance is much closer to those of ITW images than existing MoCap datasets [14, 17, 31, 37, 45], as shown in Fig. 4. Importantly, it provides accurate 3D GTs thanks to IMUs, attached to subjects' bodies and hidden under clothes. Therefore, *the 3DPW dataset serves as a bridge between MoCap*

and ITW datasets. None of previous annotation networks f [18, 23, 34] is trained on such small-scale ITW datasets with 3D GTs; instead, some of them [18, 23] are trained on additional ITW datasets with 2D GTs [1, 16]. We observed that despite its small scale (23K unique images), utilizing 3DPW as an additional training set to train the annotation network f improves the 3D pseudo-GTs of ITW datasets a lot, which results in lower 3D errors of the estimation networks g on multiple 3D benchmarks [32, 42]. On the other hand, we show that *95× larger ITW datasets (2.2M unique images [20])* with 2D GTs are not helpful for the 3D pseudo-GTs. This implies that the existence of 3D GTs in 3DPW is much more important to make 3D pseudo-GTs better than a large number of 2D GTs and rich appearance distribution from ITW datasets. Please note that the advantage of 3DPW for the annotation network f is not from the in-domain similarity between the 3DPW training and testing set. Although we use the 3DPW training set when training annotation networks f to obtain 3D pseudo-GTs, the performance of the estimation network g improves on multiple benchmarks without using 3DPW for the training of g.

2. To resolve the sub-optimality of weak supervision, initialize the annotation network f with a pre-trained 2D pose estimation network. When training the annotation network f, samples from ITW datasets are weakly supervised with 2D GTs without 3D supervision. The weak supervision might make networks converge to sub-optimal points as it involves ambiguity in nature compared to the full supervision [2, 4, 5, 9, 41, 43]. We alleviate the sub-optimality by initializing ResNet backbone [12] of our annotation network f with that of a pre-trained 2D pose estimation network [44]. From the perspective of the representation learning [10, 11, 13], the pre-trained 2D pose estimation network can extract human articulation information much better than the random initialization and ImageNet [39] classification network [12]. By extracting useful human articulation features from images at the early stage of the training, our annotation network f can reach a better convergence point, which results in more beneficial 3D pseudo-GTs.

3. To resolve the implausible articulation, use a combination of VPoser [38] and L2 regularizer in the annotation network f. When training the annotation network f, samples from ITW datasets are supervised only with 2D GTs without 3D targets (*i.e.*, 3D GTs and 3D pseudo-GTs). However, relying only on the 2D-based data term might make the networks produce 3D meshes with anatomically implausible articulations (*e.g.*, penetration and out of possible range of 3D joint rotations) as such 3D meshes can also minimize the 2D-based data term. To prevent this, we use a combination of VPoser [38] and L2 regularizer when training the annotation network f. VPoser is a variational auto-encoder, which embeds large-scale SMPL pose parameters [30] to a Gaussian latent space. It can effectively limit 3D human meshes, produced from SMPL parameters, to anatomically plausible ones. We modify our annotation network f to estimate the latent code of VPoser as the original Pose2Pose network directly estimates SMPL pose parameter. In addition, during the training, we newly apply an L2 regularizer to the estimated latent code to enforce the code to be in the latent space of VPoser.

**** Novelty of our recipes.*** Although all three recipes can be applied to the estimation network g, we observed that the effect of our recipes is much larger when they are applied to annotation networks f compared to being applied to estimation networks g. This is because annotation networks f do not have 3D evidence of ITW datasets in the training stage, while estimation networks g utilize 3D pseudo-GTs as 3D evidence of ITW datasets. Therefore, annotation networks f suffer from the three ambiguities, while estimation networks g suffer much less.

Table 1 shows a comparison of previous annotation networks and ours. Although NeuralAnnot [34] used VPoser [38] like ours, they did not investigate that the usage of VPoser is especially helpful for the annotation network f, while has a small effect when VPoser is used for the estimation network g. We show this analysis in the experimental section, which indicates that the usage of VPoser is specially designed for the annotation network f.

3. Experiment

3.1. Datasets

MoCap datsets. We use Human3.6M (H36M) [14] and MPI-INF-3DHP (MI) [31] as MoCap datasets. They are used only to train both the annotation network f and estimation network g and are not used for evaluation purposes as our goal is an evaluation on ITW benchmarks, not on MoCap ones.

ITW datasets with 2D GTs. We use COCO [28], MPII [1], and LSPET [16] as ITW datasets, which provide 2D GTs. They are used for the training of annotation networks f and estimation networks g. The inference results of annotation networks f on the above ITW datasets become 3D pseudo-GTs, used to train estimation networks g. The above ITW datasets are not used for evaluation purposes as they do not provide 3D targets.

ITW datasets with 3D GTs. We use 3DPW [42] and MuPoTS [32] as additional ITW datasets. Both contain images, captured from outdoor, with 3D GTs thanks to IMUs or multi-view marker-less motion capture systems. 3DPW training split is used to train annotation networks f and optionally estimation networks g, and 3DPW test split used to evaluate g. MuPoTS is used only for the evaluation purpose of estimation networks g.

Training sets of g	3D errors of g
H36M+MI+[COCO]$_{\text{SMPLify}}$	64.76 / 87.42
H36M+MI+[COCO]$_{\text{SMPLify-X}}$	60.40 / 81.64
H36M+MI+[COCO]$_{\text{SPIN}}$	60.70 / 80.24
H36M+MI+[COCO]$_{\text{EFT}}$	55.15 / 78.02
H36M+MI+[COCO]$_{\text{CLIFF}}$	53.36 / 75.59
H36M+MI+[COCO]$_{\text{NeuralAnnot}}$	53.34 / 76.98
H36M+MI+[COCO]$_{\text{Ours wo. first recipe}}$	50.82 / 75.63
H36M+MI+[COCO]$_{\text{Ours wo. second recipe}}$	48.84 / 75.72
H36M+MI+[COCO]$_{\text{Ours wo. third recipe}}$	48.31 / 75.70
H36M+MI+[COCO]$_{\textbf{Ours}}$	**47.52 / 74.55**

Table 2. Comparison of Pose2Pose trained with different 3D pseudo-GTs of COCO. For all settings, Pose2Pose is used as the estimation network g. The subscript at the square brackets denotes the annotation network f to obtain the 3D pseudo-GTs. The left and right 3D errors of g (PA MPJPE) are calculated on 3DPW and MuPoTS, respectively.

3.2. Evaluation protocol

As the main focus of this paper is acquiring better 3D pseudo-GTs of ITW datasets, we evaluate how much 3D pseudo-GTs are beneficial for the estimation network g. To this end, we first acquire 3D pseudo-GTs using an annotation network f. Then, we train an estimation network g using the mixed-batch training strategy, where 3D pseudo-GTs are from the annotation network f. Finally, we report the most widely used 3D error metric in the 3D human mesh estimation community, PA MPJPE, of the estimation network g on the multiple 3D ITW benchmark (*i.e.*, test split of 3DPW and MuPoTS). The errors are measured from 3D joint coordinates, extracted from 3D meshes following previous works [23, 33]. We additionally use 3DPCK as an evaluation metric of MuPoTS as previous works [8, 15]. The lower 3D errors or the higher 3DPCK of the estimation network g indicate the better 3D pseudo-GTs from the annotation network f.

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

Comparison with previous annotation networks f. Table 2 shows that Pose2Pose [33], trained with 3D pseudo-GTs of COCO from our annotation network f, achieves the lowest 3D errors on both 3DPW and MuPoTS. Even after we apply only two of three recipes, Pose2Pose trained with our f 3D pseudo-GTs still outperforms counterparts trained with 3D pseudo-GTs from previous f. For all settings, only 3D pseudo-GTs of COCO are different, and the remaining settings are the same. This proves the superiority of our annotation network f compared to previous annotation networks [18, 23, 25, 34] and iterative fitting frameworks [3, 38] regarding the ability to acquire beneficial 3D pseudo-GTs. For the comparison, we use the public 3D pseudo-GTs of previous works [18, 23, 25, 34]. The 3D pseudo-GTs of SMPLify [3] are provided in the websites of SPIN, and those of SMPLify-X [38] are obtained by running their official

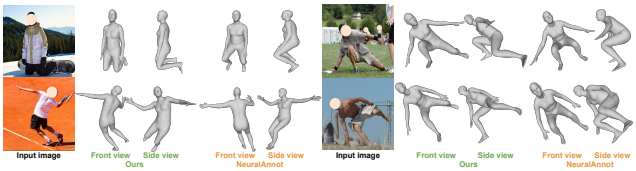

Figure 5. Visual comparison between 3D pseudo-GTs of COCO from ours and NeuralAnnot [34].

codes to 2D GT poses of COCO. Fig. 5 visually demonstrates that our 3D pseudo-GTs are better than those of NeuralAnnot [34]. NeuralAnnot fails to capture difficult poses, such as bent poses of the top-row examples. In addition, it suffers from depth ambiguity, as shown in bottom-row examples. In the bottom-left example, the right shoulder and hip should be farther from the camera than the left ones. Also, in the bottom-right example, the left leg should be behind the right leg. On the other hand, our 3D pseudo-GTs successfully capture such difficult cases.

Table 3 shows the generalizable benefits of our 3D pseudo-GTs to various state-of-the-art estimation networks g. For the experiment, we train two networks for each estimation network g using official codes of it: one with 3D pseudo-GTs of ITW datasets that it originally used, and the other with 3D pseudo-GTs of ITW datasets that are obtained from our annotation network f. Please note that other than 3D pseudo-GTs of ITW datasets, all other settings, such as the training schedule, remain the same for each estimation network g. The table shows that simply changing 3D pseudo-GTs of ITW datasets from theirs to ours greatly decreases the 3D errors. In particular, the error of the z-axis decreases the most among x-, y- and z-axis errors, which shows that our 3D pseudo-GTs effectively alleviate the depth ambiguity of the 3D human mesh estimation from a monocular image. The reason for the relatively small z-axis error gap of METRO [26] is that it is additionally trained on 3DPW. Nevertheless, our 3D pseudo-GTs still enhance its performance. As detailed training set configurations of PARE [22] are not publicly available, we simply trained the PARE network only on COCO, the reason for different 3D errors from their paper.

Pushing the performance of state-of-the-art networks. Using our 3D pseudo-GTs of ITW datasets, we investigate how far state-of-the-art networks can become better. To this end, we re-trained 3DCrowdNet [8] with our 3D pseudo-GTs and stretched its training schedule two times. Table 4 and 5 show that our 3DCrowdNet outperforms all existing methods on both 3DPW and MuPoTS. In Table 4, for the fair comparison with recent works [22, 26, 27] that use 3DPW to train their networks, we additionally show our result when 3DCrowdNet is additionally trained on 3DPW. Please note that 3DCrowdNet with its original 3D pseudo-GTs and stretched schedule produces a 50.1 3D error, much

Estimation networks g	Training sets of g	3D errors of g
SPIN [23]	H36M+MI+[COCO+MPII+LSPET]$_{\text{SMPLify}}$	59.6 (21.6/24.2/40.8)
	H36M+MI+[COCO+MPII+LSPET]$_{\textbf{Ours}}$	**51.6 (19.0/20.7/35.4)**
I2L-MeshNet [35]	H36M+MuCo+[COCO]$_{\text{SMPLify-X}}$	57.7 (20.6/21.7/40.8)
	H36M+MuCo+[COCO]$_{\textbf{Ours}}$	**47.1 (17.1/18.5/32.6)**
Pose2Pose [33]	H36M+[COCO+MPII]$_{\text{NeuralAnnot}}$	54.4 (19.1/20.3/39.0)
	H36M+[COCO+MPII]$_{\textbf{Ours}}$	**49.6 (18.4/18.8/34.7)**
3DCrowdNet [8]	H36M+MuCo+CrowdPose+[COCO+MPII]$_{\text{NeuralAnnot}}$	51.5 (17.6/18.2/36.3)
	H36M+MuCo+CrowdPose+[COCO+MPII]$_{\textbf{Ours}}$	**47.2 (16.8/17.7/33.5)**
PARE [22]	[COCO]$_{\text{EFT}}$	57.3 (20.3/20.3/41.7)
	[COCO]$_{\textbf{Ours}}$	**47.3 (17.5/18.2/32.9)**
PyMAF [46]	H36M+MI+[COCO+MPII+LSPET]$_{\text{SPIN}}$	58.9 (21.0/23.7/41.8)
	H36M+MI+[COCO+MPII+LSPET]$_{\textbf{Ours}}$	**50.9 (18.0/20.8/35.1)**
METRO [26]	H36M+UP3D+MuCo+3DPW+MPII+[COCO]$_{\text{SMPLify-X}}$	47.9 (18.8/18.5/32.4)
	H36M+UP3D+MuCo+3DPW+MPII+[COCO]$_{\textbf{Ours}}$	**45.8 (17.9/17.2/31.3)**

Table 3. Comparison of various estimation networks g, trained with different 3D pseudo-GTs of ITW datasets. Notations are the same as Table 2, except the 3D errors of g (PA MPJPE) are calculated on 3DPW, and three errors in the parenthesis are from x-, y-, and z-axis, respectively.

Estimation networks g	3D errors of g
SPIN [23]	59.2
Pose2Mesh [7]	58.9
PyMAF [46]	58.9
I2L-MeshNet [35]	57.7
Pose2Pose [33]	54.4
ROMP [40]	53.3
3DCrowdNet [8]	51.5
PARE [22]	49.3
HybrIK [24]	48.8
Our 3DCrowdNet	**46.1**
METRO* [26]	47.9
PARE* [22]	46.4
MeshGraphormer* [27]	45.6
Our 3DCrowdNet*	**43.6**

Table 4. Comparisons of 3D human mesh estimation methods on 3DPW. * denotes additional training on 3DPW.

Estimation networks g	3DPCK of g	
	All	Matched
SMPLify-X [38]	62.8	68.0
HMR [19]	66.0	70.9
Jiang et al. [15]	69.1	72.2
3DCrowdNet [8]	72.7	73.3
Our 3DCrowdNet	**76.2**	**76.9**

Table 5. Comparisons of 3D human mesh estimation methods on MuPoTS. The higher the better.

worse than our 46.1 3D error on 3DPW. The tables show the power of our 3D pseudo-GTs, which elevate a state-of-the-art estimation network g to a top-performing method.

3.4. Ablation study

For all the ablation studies, our annotation network f produces 3D pseudo-GTs of COCO. Then, the 3D pseudo-GTs of COCO in addition to H36M, MI, and optionally 3DPW are used to train the estimation network g. We use Pose2Pose [33] for our estimation network g.

First stage	Second stage	3D errors
Annot. network f	Annot. network f	48.21 / 75.55
	Fine-tuned annot. network f	47.93 / 75.13
	Est. network g	**46.21 / 74.40**

Table 6. Comparison of 3D errors (PA MPJPE) of various pipelines on 3DPW. The left and right 3D errors of g (PA MPJPE) are calculated on 3DPW and MuPoTS, respectively.

Justification of using separated networks in our two-stage framework. As shown in Fig. 2, our framework for the 3D human mesh estimation in the wild consists of two stages, of which the first stage is obtaining 3D pseudo-GTs with annotation network f, and the second stage is training estimation network g with the 3D pseudo-GTs of the first stage. The annotation network f and estimation network g are separated. To justify using separated networks in our two-stage framework, we compare two variants with our setting in Table 6. The first variant is testing the annotation network f on 3DPW. Although the purpose of the annotation network is to obtain 3D pseudo-GTs of ITW datasets, we can use it as an estimation network and test it on 3DPW. The second variant measures the 3D error using the annotation network f after fine-tuning it with the 3D pseudo-GTs. This setting uses 3D pseudo-GTs from the annotation network f (the first variant) for the fine-tuning; however, it uses one network instead of the separated two networks like ours. Finally, our estimation network g is trained with the 3D pseudo-GTs, where the 3D pseudo-GTs are from the annotation network f (the first variant). For a fair comparison, all networks in the table have the same network architecture of Pose2Pose [33]. The table shows that our estimation network g achieves the lowest error, which justifies using separated networks in our two-stage framework. The reason for the better performance of our setting is that it is *fully supervised* with 3D targets (*i.e.*, 3D pseudo-GTs) from the start of

Recipes	Where to apply recipe	3D errors of g
Train on 3DPW	None	50.82 / 75.63
	Annotation network f	47.13 / 74.43
	Estimation network g	48.33 / 74.84
	Both f and g	45.98 / 73.97
Initialize with 2D pose network	None	48.84 / 75.72
	Annotation network f	46.99 / 74.58
	Estimation network g	48.13 / 74.73
	Both f and g	45.98 / 73.97
Use VPoser and L2 reg.	None	48.31 / 75.70
	Annotation network f	46.21 / 74.40
	Estimation network g	48.13 / 75.72
	Both f and g	45.98 / 73.97

Table 7. Comparison of 3D errors of estimation networks g, trained with different settings. The left and right 3D errors of g (PA MPJPE) are calculated on 3DPW and MuPoTS, respectively.

	Annotation network f		Estimation network g	
ID	Training sets	Unique images	Training sets	3D errors
$f1$	H36M+MI+COCO	919K	H36M+MI+[COCO]$_{f1}$	53.02 / 77.04
$f2$	H36M+MI+COCO+MPII	919K+29K	H36M+MI+[COCO]$_{f2}$	53.23 / 77.05
$f3$	H36M+MI+COCO+LSPET	919K+9K	H36M+MI+[COCO]$_{f3}$	54.14 / 77.53
$f4$	H36M+MI+COCO+InstaVariety	919K+2185K	H36M+MI+[COCO]$_{f4}$	53.86 / 78.49
$f5$	H36M+MI+COCO+3DPW	919K+23K	H36M+MI+[COCO]$_{f5}$	**51.61 / 75.37**

Table 8. Comparison of 3D errors of estimation networks g, trained with different 3D pseudo-GTs of COCO. Notations are the same as Table 2.

the training. On the other hand, the two variants are *weakly supervised* with 2D targets (*i.e.*, 2D GT poses) from the start of the training, although the second variant is fine-tuned with 3D targets later. The weak supervision at the start of the training makes the two variants converge to sub-optimal points compared to the estimation network g. All networks in the table are trained on H36M+MI+MSCOCO+3DPW. In addition, ResNet [12] of them are initialized with pre-trained 2D pose estimation network [44].

Applying the recipes to estimation networks g. Table 7 shows that applying our recipes to annotation network f improves the 3D errors on both benchmarks much more than applying them to estimation networks g. Applying the recipes to both annotation network f and estimation network g performs the best; however, the performance improvement is limited compared to the improvement brought by applying the recipes only to annotation network f. For example, applying each recipe to annotation network f brings 3.69, 1.85, and 2.1 3D error improvement, respectively, while applying additionally to both f and g brings 1.15, 1.01, and 0.23 3D error improvement. This shows that our recipes are specially designed for the annotation networks f to obtain beneficial 3D pseudo-GTs. The reason for the small effect when the recipes are applied to the estimation networks g is that the estimation networks g are trained with 3D pseudo-GTs, while annotation networks f are trained with 2D GTs of ITW datasets without 3D evidence. The absence of 3D evidence when training annotation networks f results in severe ambiguities, which can be cured by our recipes. On the other hand, as estimation networks g are fully supervised with 3D pseudo-GTs, they suffer less from ambiguities. For each recipe, *None* represents both annotation and estimation networks are trained with the remaining other two recipes.

Effect of training annotation network f on 3DPW. Table 8 shows how 3DPW changes the 3D pseudo-GTs compared to other ITW datasets, such as MPII, LSPET, and InstaVariety [20]. As the table shows, adding other ITW datasets does not obtain the performance gain of g compared to H36M+MI+COCO. This is because adding ITW datasets does not contribute to relieving the depth ambiguity as they provide only 2D GTs. On the other hand, 3DPW provides 3D GTs, largely helpful to alleviate the depth ambiguity. Importantly, the 3D errors of g on MuPoTS decrease as well, which implies that using 3DPW as an additional training set is beneficial for multiple 3D ITW benchmarks.

It is noticeable that InstaVariety has 95 times more images than 3DPW, while much less helpful for the beneficial 3D pseudo-GTs. This tells us that for the 3D pseudo-GTs of ITW datasets, the existence of 3D GTs is much more important than a large number of 2D GTs and rich appearance distribution from ITW datasets. It suggests a different research direction compared to recent representation learning methods [10, 11, 13] as they suggest that collecting large-scale unlabeled images can boost the image classification performance a lot. Our analysis is consistent with Table 9. The table shows that when we only use 2D GTs of 3DPW with-

ID	Annotation network f Training sets	Estimation network g Training sets	3D errors
$f1$	H36M+MI+COCO	H36M+MI+[COCO]$_{f1}$	53.02 (18.7/19.4/38.1)
$f2$	H36M+MI+COCO+3DPW without 3D GTs	H36M+MI+[COCO]$_{f2}$	53.66 (18.8/19.6/38.6)
$f3$	H36M+MI+COCO+3DPW	H36M+MI+[COCO]$_{f3}$	**51.61 (18.4/19.2/36.9)**

Table 9. Comparison of 3D errors of estimation networks g, trained with different 3D pseudo-GTs of COCO. Notations are the same as Table 3.

ID	Annotation network f Initialization	Training sets	Estimation network g Training sets	3D errors
$f1$	ImageNet cls. [12]		H36M+MI+[COCO]$_{f1}$	51.61 (18.4/19.2/36.9)
$f2$	3D pose [23]	H36M+MI+COCO+3DPW	H36M+MI+[COCO]$_{f2}$	51.62 (18.4/19.0/37.0)
$f3$	2D pose [44]		H36M+MI+[COCO]$_{f3}$	**47.52 (17.4/18.4/33.0)**

Table 10. Comparison of 3D errors of estimation networks g, trained with different 3D pseudo-GTs of COCO. Notations are the same as Table 3.

out 3D GTs, the quality of 3D pseudo-GTs does not change much, which leads to similar 3D errors of g compared to H36M+MI+COCO. The result shows that the performance gain from using 3DPW is not from images of 3DPW, but from 3D GTs of 3DPW. In particular, most of the performance gain is from the z-axis, which shows the effectiveness of using 3DPW to resolve the depth ambiguity.

Effect of initializing annotation network f with a pre-trained 2D pose network. Table 10 shows that initializing annotation network f with a pre-trained 2D pose network [44] produces more beneficial 3D pseudo-GTs, which result in lower 3D errors of g compared to the conventional ImageNet classification pre-training [12]. This is because initializing with the pre-trained 2D pose network makes the annotation network f extract useful human articulation features from images at the early stage of the training. Therefore, better initialization results in a better convergence point, which alleviates the sub-optimality of weak supervision. Interestingly, the z-axis error decreases much, while the errors of x- and y-axis remain similar. This indicates that the proposed initialization does not simply result in better 2D pose estimation ability, but helps our annotation network f to produce more beneficial 3D pseudo-GTs. We further compare our initialization with the initialization of EFT [18], which initializes their network with pre-trained 3D pose estimation network [23]. We observed that initializing the network with pre-trained 3D pose estimation network [23] produces almost the same results as the ImageNet counterpart and is largely beaten by our 2D-based initialization. We think this is because the pre-trained 3D pose network [23] is already converged to produce lower quality 3D pseudo-GTs than ours.

4. Related works

SMPLify [3] and SMPLify-X [38] are iterative fitting frameworks, which iteratively fit SMPL parameters to target 2D pose by minimizing energy functions. Using them to 2D GT pose of ITW datasets, researchers [7, 26, 27, 35] obtained 3D pseudo-GTs. Recently, several annotation networks are introduced. SPIN [23] predicts SMPL parameters using a network and iteratively fits [3] the predicted parameters to 2D GT pose. Their final 3D pseudo-GT of each sample is obtained by selecting one with smaller SMPLify loss [3] between their fit and prepared initial 3D pseudo-GT. The initial 3D pseudo-GTs are prepared before training their network by running SMPLify [3] to 2D GT pose. The final 3D pseudo-GTs are used to train an HMR [19] regressor. EFT [18] fine-tunes the pre-trained SPIN to the 2D GT pose of each sample, and the outputs of the last fine-tuning iteration become the 3D pseudo-GT of the sample. Both SPIN and EFT require initial 3D pseudo-GTs from SMPLify [3] to train their networks. On the other hand, NeuralAnnot [34] is weakly supervised with 2D GT pose without requiring initial 3D pseudo-GTs. Compare to them, our annotation network produces more beneficial 3D pseudo-GTs, which results in much lower 3D errors of the estimation networks (Table 2 and 3). Table 1 shows differences between our annotation networks and the above ones.

5. Conclusion

We introduce three recipes to obtain highly beneficial 3D pseudo-GTs of ITW datasets for the 3D human mesh estimation in the wild. Experimental results show that simply re-training state-of-the-art networks with our 3D pseudo-GTs elevates their performance to the next level. In addition, we show our 3D pseudo-GTs are much more beneficial than previous ones. In closing, we hope the community to have more remarks on the importance of 3D pseudo-GTs.

References

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 2014. 1, 4

[2] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. 4

[3] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *ECCV*, 2016. 2, 5, 8

[4] Junsuk Choe, Seong Joon Oh, Sanghyuk Chun, Seungho Lee, Zeynep Akata, and Hyunjung Shim. Evaluation for weakly supervised object localization: Protocol, metrics, and datasets. *TPAMI*, 2022. 4

[5] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *CVPR*, 2020. 4

[6] Hongsuk Choi, Gyeongsik Moon, Ju Yong Chang, and Kyoung Mu Lee. Beyond static features for temporally consistent 3D human pose and shape from a video. In *CVPR*, 2021. 1, 2

[7] Hongsuk Choi, Gyeongsik Moon, and Kyoung Mu Lee. Pose2Mesh: Graph convolutional network for 3D human pose and mesh recovery from a 2D human pose. In *ECCV*, 2020. 1, 2, 6, 8

[8] Hongsuk Choi, Gyeongsik Moon, JoonKyu Park, and Kyoung Mu Lee. Learning to estimate robust 3D human mesh from in-the-wild crowded scenes. In *CVPR*, 2022. 1, 2, 3, 5, 6

[9] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord. WILDCAT: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *CVPR*, 2017. 4

[10] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020. 4, 7

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 4, 7

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 4, 7, 8

[13] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. In *ICCV*, 2021. 4, 7

[14] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *TPAMI*, 2014. 1, 3, 4

[15] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *CVPR*, 2020. 5, 6

[16] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. In *CVPR*, 2011. 1, 4

[17] Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic Studio: A massively multiview system for social motion capture. In *ICCV*, 2015. 1, 3

[18] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3D human pose fitting towards in-the-wild 3D human pose estimation. In *3DV*, 2021. 2, 3, 4, 5, 8

[19] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018. 6, 8

[20] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3D human dynamics from video. In *CVPR*, 2019. 4, 7

[21] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. VIBE: Video inference for human body pose and shape estimation. In *CVPR*, 2020. 1, 2

[22] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. PARE: Part attention regressor for 3D human body estimation. In *ICCV*, 2021. 1, 2, 3, 5, 6

[23] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In *ICCV*, 2019. 2, 3, 4, 5, 6, 8

[24] Jiefeng Li, Chao Xu, Zhicun Chen, Siyuan Bian, Lixin Yang, and Cewu Lu. HybrIK: A hybrid analytical-neural inverse kinematics solution for 3D human pose and shape estimation. In *CVPR*, 2021. 6

[25] Zhihao Li, Jianzhuang Liu, Zhensong Zhang, Songcen Xu, and Youliang Yan. CLIFF: Carrying location information in full frames into human pose and shape estimation. In *ECCV*, 2022. 5

[26] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 2021. 1, 2, 3, 5, 6, 8

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

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 3, 4

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

[30] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. AMASS: Archive of motion capture as surface shapes. In *ICCV*, 2019. 4

[31] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3D human pose estimation in the wild using improved CNN supervision. In *3DV*, 2017. 1, 3, 4

[32] Dushyant Mehta, Oleksandr Sotnychenko, Franziska Mueller, Weipeng Xu, Srinath Sridhar, Gerard Pons-Moll, and Christian Theobalt. Single-shot multi-person 3D pose estimation from monocular RGB. In *3DV*, 2018. 4

[33] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. Accurate 3D hand pose estimation for whole-body 3D human mesh estimation. In *CVPRW*, 2022. 1, 2, 3, 5, 6

[34] Gyeongsik Moon, Hongsuk Choi, and Kyoung Mu Lee. NeuralAnnot: Neural annotator for 3D human mesh training sets. In *CVPRW*, 2022. 2, 3, 4, 5, 8

[35] Gyeongsik Moon and Kyoung Mu Lee. I2L-MeshNet: Image-to-Lixel prediction network for accurate 3D human pose and mesh estimation from a single RGB image. In *ECCV*, 2020. 1, 2, 3, 6, 8

[36] Gyeongsik Moon, Hyeongjin Nam, Takaaki Shiratori, and Kyoung Mu Lee. 3D clothed human reconstruction in the wild. In *ECCV*, 2022. 2

[37] Gyeongsik Moon, Shoou-I Yu, He Wen, Takaaki Shiratori, and Kyoung Mu Lee. InterHand2.6M: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image. In *ECCV*, 2020. 1, 3

[38] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *CVPR*, 2019. 2, 4, 5, 6, 8

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *IJCV*, 2015. 4

[40] Yu Sun, Qian Bao, Wu Liu, Yili Fu, Black Michael J., and Tao Mei. Monocular, one-stage, regression of multiple 3D people. In *ICCV*, 2021. 1, 2, 6

[41] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Yuille. PCL: Proposal cluster learning for weakly supervised object detection. *TPAMI*, 2018. 4

[42] Timo von Marcard, Roberto Henschel, Michael J Black, Bodo Rosenhahn, and Gerard Pons-Moll. Recovering accurate 3D human pose in the wild using IMUs and a moving camera. In *ECCV*, 2018. 1, 2, 3, 4

[43] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-MIL: Continuation multiple instance learning for weakly supervised object detection. In *CVPR*, 2019. 4

[44] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. In *ECCV*, 2018. 3, 4, 7, 8

[45] Zhixuan Yu, Jae Shin Yoon, In Kyu Lee, Prashanth Venkatesh, Jaesik Park, Jihun Yu, and Hyun Soo Park. HUMBI: A large multiview dataset of human body expressions. In *CVPR*, 2020. 1, 3

[46] Hongwen Zhang, Yating Tian, Xinchi Zhou, Wanli Ouyang, Yebin Liu, Limin Wang, and Zhenan Sun. PyMAF: 3D human pose and shape regression with pyramidal mesh alignment feedback loop. In *ICCV*, 2021. 6