Exemplar Fine-Tuning for 3D Human Model Fitting Towards In-the-Wild 3D Human Pose Estimation

Hanbyul Joo **Natalia Neverova** **Andrea Vedaldi**
Facebook AI Research

Abstract

We present Exemplar Fine-Tuning (EFT), a new method to fit a 3D parametric human model to a single RGB input image cropped around a person with 2D keypoint annotations. While existing parametric human model fitting approaches, such as SMPLify, rely on the "view-agnostic" human pose priors to enforce the output in a plausible 3D pose space, EFT exploits the pose prior that comes from the specific 2D input observations by leveraging a fully-trained 3D pose regressor. We thoroughly compare our EFT with SMPLify, and demonstrate that EFT produces more reliable and accurate 3D human fitting outputs on the same inputs. Especially, we use our EFT to augment a large scale in-the-wild 2D keypoint datasets, such as COCO and MPII, with plausible and convincing 3D pose fitting outputs. We demonstrate that the pseudo ground-truth 3D pose data by EFT can supervise a strong 3D pose estimator that outperforms the previous state-of-the-art in the standard outdoor benchmark (3DPW), even without using any ground-truth 3D human pose datasets such as Human3.6M. Our code and data are available at `https://github.com/facebookresearch/eft`.

1 Introduction

We consider the problem of reconstructing the pose of humans in 3D from single 2D images, a key task in applications such as human action recognition, human-machine interaction and virtual and augmented reality. Since individual 2D images do not contain sufficient information for 3D reconstruction, algorithms must supplement the missing information by a learned prior on the plausible 3D poses of the human body. Established approaches such as SMPLify [1, 2] cast this as fitting the parameters of a 3D human model [3, 4, 5] to the location of 2D keypoints, manually or automatically annotated in images. They regularize the solution by means of a "view-agnostic" 3D pose prior, which incurs some limitations: fitting ignores the RGB images themselves, the 3D priors, often learned separately in laboratory conditions, lack realism, and balancing between the prior and the data term (*e.g.*, 2D keypoint error) is difficult.

In this paper, we present **Exemplar Fine-Tuning** (EFT), a new technique to fit 3D parametric models to 2D annotations which overcomes these limitations and results in much better 3D reconstructions. The idea of EFT is to start from an image-based 3D pose regressor such as [6, 7, 8]. The pretrained regressor *implicitly* embodies a strong pose prior *conditional* on the observation of the RGB input image, providing a function $\Theta = \Phi_{\mathbf{w}}(\mathbf{I})$ sending the image $\mathbf{I}$ to the parameters Θ of the 3D body. Our intuition is to look at this function as a *conditional re-parameterization* of pose, where the conditioning factor is the image $\mathbf{I}$, and the new parameters are the weights $\mathbf{w}$ of the underlying (neural network) regressor. We show that, when this re-parameterization is used, fitting 2D keypoints results in better 3D reconstructions than using existing fitting methods such as SMPLify, while also improving the output of the regressor Φ itself.

The name of our technique is justified by the fact that fitting the parameters $\mathbf{w}$ of the predictor to a 2D example is similar to performing a training step for the regressor Φ. However, this is only done for

Figure 1: **Exemplar Fine-Tuning** (EFT) fits a parametric 3D human model given as input a single RGB image and 2D keypoint annotations. (Column 1) Input image with 2D keypoint annotations; (Columns 2-4) EFT iterations; (Column 5) a side view; (Column 6-7) output of SMPLify [1] without using a 3D pose priors; (Column 8-9) SMPLify [1] with 3D pose priors. For SMPLify, a staged technique is used to avoid local minima (100 iterations in total), while EFT optimizes all parts together without using any external 3D pose priors.

the purpose of reconstructing a particular exemplar, not for learning the network. In fact, the updated parameters are discarded before processing the next sample.

We show two important application of this technique. The first is 'direct': for in-the-wild data, EFT results in better single-view 3D body pose reconstruction than any existing approach [9, 6, 10, 11, 12, 13, 14, 7, 15, 16, 17], and thus can be used as a drop-in replacement of these techniques in current applications. As shown in Fig. 1, EFT fits are particularly robust to challenging 2D poses than defy traditional fitting and regression methods [3, 5, 4, 1, 2]. This is possible because EFT can leverage the implicit prior learned by the neural network regressor. Furthermore, this prior is conditioned on the specific the RGB input image, which contains more information than the 2D keypoint locations. At the same time, it results in more accurate 2D fits than the regressor alone.

The second application of EFT is to generate *3D body pose data in the wild*. We show this by taking an existing large-scale 2D dataset such as COCO [18] and using EFT to augment it with approximate 3D body pose annotations. Remarkably, we show that existing supervised pose regressor methods can be trained using these pseudo-3D annotations as well or better than using ground truth 3D annotations in existing 3D datasets [19, 20, 21], which, for the most part, are collected in laboratory condition. In fact, we show that our 3D-fied in the wild data, which we will release to the public, is *sufficient* to train state-of-the-art 3D pose regressors by itself, outperforming methods trained on the combination of datasets with 3D and 2D ground-truth [7, 22, 8].

2 Related Work

Deep learning has significantly advanced 2D pose recognition [23, 24, 25, 26, 27, 28], facilitating the more challenging task of 3D reconstruction [9, 6, 10, 11, 12, 13, 14, 7, 15, 17, 16, 22], our focus.

Single-image 3D human pose estimation. Single-view 3D pose reconstruction methods differ in how they incorporate a 3D pose prior and in how they perform the prediction. *Fitting-based methods* assume a 3D body model such as SMPL [3] and SCAPE [29], and use an optimization algorithm to fit it to the 2D observations. While early approaches [30, 31] required manual input, starting with SMPLify [1] the process has been fully automatized, then improved in [2] to use silhouette annotations, and eventually extended to multiple views and multiple people [32]. *Regression-based methods*, on the other hand, predict 3D pose directly. The work of [33] uses sparse linear regression that incorporates a tractable but somewhat weak pose prior. Later approaches use instead deep neural networks, and differ mainly in the nature of their inputs and outputs [9, 6, 10, 11, 12, 13, 14, 7, 15, 17, 16, 34, 35]. Some works start from a pre-detected 2D skeleton [10] while others start from raw images [7]. Using a 2D skeleton relies on the quality of the underlying 2D keypoint detector and discards appearance details that could help fitting the 3D model to the image. Using raw images can potentially make use of this information, but training such models from current 3D indoor datasets might fail to generalize to unconstrained images. Hence several papers combine 3D indoor datasets with 2D in-the-wild ones [7, 17, 15, 6, 14, 13, 7, 35, 22]. Methods also differ in their output, with some predicting 3D keypoints directly [10], some predicting the parameters of a 3D human body model [7, 17], and others volumetric heatmaps for the body joints [11]. Finally, *hybrid methods* such as SPIN [5] or MTC [17] combine fitting and regression approaches.

3D reconstruction without paired 3D ground-truth. While regression methods usually require image datasets paired with 3D ground truth annotations, fitting methods such as SMPLify rely only on 2D annotations by predicting a small number of model parameters and by using a prior learned on independent motion capture data. However, their output quality is largely dependent on the initialization, with problematic results for challenging poses (*e.g.*, see Fig. 2 left panel). Furthermore, the space of plausible human body poses can be described empirically, by collecting a large number of samples in laboratory conditions [36, 37, 5], but this may lack realism. Regression methods [7, 38] can also be learned without requiring images with 3D annotations, by combining 2D datasets with a parametric 3D model *and* empirical motion capture pose samples, integrating them into their neural network regressor by means of adversarial training. However, while the predictions obtained by such methods are plausible, they often do not fit the 2D data very accurately. Fitting could be improved by refining this initial solution by means of an optimization-based method as in SPIN [8], but empirically we have found that this distorts the pose once again, leading to solutions that are not plausible anymore.

Human pose datasets. There are several in-the-wild datasets with sparse 2D pose annotations, including COCO [18], MPII [39], Leeds Sports Pose Dataset (LSP) [40, 41], PennAction [42] and Posetrack [43]. Furthermore, Dense Pose [44] has introduced a dataset with dense surface point annotations, mapping images to a UV representation of a parametric 3D human model [3]. Compared to annotating 2D keypoints, annotating 3D human poses is much more challenging as there are no easy-to-use or intuitive tools to input the annotations. Hence, current 3D annotations are mostly obtained by means of motion capture systems in indoor environments. Examples include the Human3.6M dataset [19], Human Eva [45], Panoptic Studio [21], and MPI-INF-3DHP [46]. These datasets provide 3D motion capture data paired with 2D images, but the images are very controlled. 3DPW dataset [51] is exceptional by capturing outdoor scenes by a hand-held video camera and IMUs. There exists an approach to produce a dataset with 3D pose annotations on Internet photos [2]. However, the traditional optimization-based fitting method used in this work limits the quality and size of dataset. There are also several large scale motion capture datasets that do not have corresponding images at all (*e.g.* CMU Mocap [47] and KIT [48]). These motion capture datasets have recently been reissued in a unified format in the AMASS dataset [49].

3 Method

Parametric human models [29, 3, 4, 5] represent human shapes and poses by means of a small number of parameters, while capturing constraints such as symmetry and limb proportions. Here, we focus as an example the SMPL model [3], although any other model could be used instead. The SMPL parameters $\Theta = (\boldsymbol{\theta}, \boldsymbol{\beta})$ comprise the pose parameters $\boldsymbol{\theta} \in \mathbb{R}^{24\times 3}$, which control the rotations of 24 body joints with respect to their parent joints, and the shape parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$, which control the shape of the body by means of 10 principal directions of variations. The 3D location $\mathbf{J} \in \mathbb{R}^{24\times 3}$ of the body joints is obtained by first finding their configuration at rest using the shape parameters $\boldsymbol{\beta}$, and then by applying the joint rotations Θ following the skeletal hierarchy. SMPL also includes a mesh component that deforms along with the skeleton, but we ignore it here as the major loss constrains only the 3D location $\mathbf{J}$ of the joints. Hence, for us SMPL reduces to a function $\mathbf{J} = M(\Theta)$.

Given an image $\mathbf{I}$ of a human, the goal then is to find the parameters Θ of SMPL that match the pose of the subject. **Fitting-based approaches** [1, 2] take 2D geometric cues such as joints, silhouettes, and part labels, and optimizes the model parameters Θ to fit the 3D model to the 2D cues. For example, assume that the 2D locations $\hat{\mathbf{j}} \in \mathbb{R}^{24\times 2}$ of the body joints are given. Furthermore, let $\pi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ be the *camera projection function*, mapping 3D points to their 2D image locations. Then, reconstructing the 3D joints $\mathbf{J}$ from the 2D cues $\hat{\mathbf{j}}$ amouts to fitting the SMPL model:

$$\mathbf{J}(\hat{\mathbf{j}}) = M(\Theta^*) \quad \text{where} \quad (\Theta^*, \pi^*) = \underset{\Theta,\pi}{\operatorname{argmin}} L_{\text{2D}}\left(\pi\left(M(\Theta)\right), \hat{\mathbf{j}}\right) + L_{\text{prior}}\left(\Theta\right), \tag{1}$$

where the data term L_{2D} is the re-projection error between reconstructed and observed 2D keypoints. Note that, since the viewpoint of the image is unknown, the camera parameters π must be optimized together with the body parameters Θ.

Due to the lack of depth information, optimizing L_{2D} is insufficient to recover the pose parameters Θ uniquely; to remove this ambiguity, one adds a prior term L_{prior} to the loss in order to prioritize plausible solutions. SMPLify [1] expresses this prior via a mixture of Gaussians or naïve thresholding, using a separate 3D motion capture dataset [47, 37, 49] to learn the prior parameters beforehand.

Dataset	COCO-Part	COCO-All	MPII	LSP	H36M	MPI-INF	PanopticDB	3DPW (Train.)
Sample Num.	28K	79K	14K	8K	312K	96K	736K	22K

Table 1: Summary of public DBs: 3DPW [51], COCO [18], H36M [19], MPII [39], MPI-INF [20], Panoptic Studio [21, 17], LSPet [40, 41]. We use our EFT to generate the pseudo 3D poses for all 2D pose DBs.

A disadvantage of fitting methods is that the objective (1) can usually be optimized only locally, e.g. via a gradient descent method, so success depends strongly on the quality of the initialization [1, 2]. This requires ad-hoc steps to avoid bad local minima, such as first aligning the torso and then optimizing the limbs. Furthermore, in (1) it is difficult to balance the re-projection loss with the prior: too strong a prior may force the model to output a "mean" pose ignoring the 2D evidence, and too weak a prior may result in implausible or distorted poses.

By contrast, **regression-based approaches** predict the SMPL parameters Θ directly from geometric and non-geometric 2D cues $\mathbf{I}$, including raw images [7] and sparse [10] and dense [34] keypoints. The mapping is implemented by a neural network $\Theta = \Phi_{\mathbf{w}}(\mathbf{I})$ trained on a combination of indoor datasets [19, 20] with 3D ground-truth and in-the-wild ones [18, 39] with only 2D annotations. This is done by optimizing

$$\mathbf{w}^* = \underset{\Phi}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} L_{\text{2D}} \left(\pi \left(M(\Phi_{\mathbf{w}}(\mathbf{I}_i)) \right), \hat{\mathbf{j}}_i \right) + \mu_i L_{\mathbf{J}} \left(M(\Phi_{\mathbf{w}}(\mathbf{I}_i)), \hat{\mathbf{J}}_i \right) + \tau_i L_{\Theta} \left(\Phi_{\mathbf{w}}(\mathbf{I}_i), \hat{\Theta}_i \right), \tag{2}$$

adding to the 2D re-projection loss L_{2D} the reconstruction losses $L_{\mathbf{J}}$ and L_{Θ} for the 3D joints and SMPL parameters, where $\hat{\mathbf{j}}_i$, $\hat{\mathbf{J}}_i$ and $\hat{\Theta}_i$ are, respectively, the ground truth 2D joints, 3D joints, and SMPL parameters for the i-th training samples $\mathbf{I}_i$. μ_i and τ_i are loss-balancing coefficients and they can be set to zero for samples that do not have 3D annotations. The parameters for the camera projection function π can be predicted as additional outputs of the neural network Φ [7, 50, 34, 8].

During training, the regressor $\Phi_{\mathbf{w}^*}$ acquires an implicit prior on possible human poses, optimized for 3D reconstruction. This is arguably stronger than the prior L_{prior} used by fitting methods, which are learned separately, on different 3D data, and tend to regress to a mean solution. On the other hand, while both fitting and regression methods aim at minimizing the same 2D reprojection error L_{2D}, fitting methods do so explicitly for each sample, obtaining more accurate 2D fits, while regression methods minimize this loss only at training time.

In order to combine the advantages of both, **Exemplar Fine-Tuning** (EFT), our approach, interprets the network Φ as a re-parameterization $\mathbf{J} = M(\Theta) = M(\Phi_{\mathbf{w}}(\mathbf{I}))$ of the 3D joints $\mathbf{J}$ as a function of the network parameters $\mathbf{w}$ (instead of the SMPL parameters Θ).With this, we can rewrite Eq. (1) as:

$$\mathbf{J}(\hat{\mathbf{j}}, \mathbf{I}) = M(\Phi_{\mathbf{w}^+}(\mathbf{I})) \quad \text{where} \quad \mathbf{w}^+ = \underset{\mathbf{w} \in \mathcal{N}(\mathbf{w}^*)}{\operatorname{argmin}} L_{\text{2D}} \left(\pi \left(M(\Phi_{\mathbf{w}}(\mathbf{I})) \right), \hat{\mathbf{j}} \right) + \lambda \|\boldsymbol{\beta}\|_2^2, \tag{3}$$

Note that we dropped the prior L_{prior} as one is already implicitly captured by the network (this also removes the difficult problem of balancing L_{prior} and L_{2D} discussed above). Eq. (3) indicates that the optimization is carried out in a neighborhood of the pretrained parameters $\mathbf{w}^*$ from Eq. (2). In practice, this is done implicitly, by using the same local optimizer used for the learning objective (2) starting from $\mathbf{w}^*$. However, the goal is *not* to learn/improve the network, but rather to estimate the pose of a single sample. In fact, $\mathbf{w}$ is re-initialized to $\mathbf{w}^*$ for each new sample before optimizing Eq. (3). Compared to the traditional fitting methods (1), EFT is able to leverage both the 2D joints $\hat{\mathbf{j}}$ and the RGB image $\mathbf{I}$ for estimation. Furthermore, $\Phi_{\mathbf{w}}(\mathbf{I})$ provides a prior which is tuned to the input $\mathbf{I}$, which is less prone to regressing a mean pose compared to the input-agnostic prior L_{prior} used by other fitting methods. Compared to the regression methods, EFT *maintains the plausibility* of the initially regressed pose, but reduces the 2D re-projection errors. This is the case even for samples that, due to occlusions or unusual poses, challenge the regressor.

Previous method	3DPW ↓	H36M ↓
HMR [7]	81.3	56.8
DSD [22]	75.0	44.3
HoloPose (w/ post-proc.) [52]	—	46.5
SPIN [8]	59.2	41.1
HMR (temporal) [50]	72.6	56.9
VIBE (temporal) [35]	56.5	41.4
VIBE (temporal) (w/ 3DPW train) [35]	51.9	41.5
Direct 3D supervision, 3D pseudo-ground truth from EFT		
H36M	146.6	54.9
PanopticDB (PanDB)	115.4	107.9
MPI-INF-3DHP (MI)	97.3	106.2
3DPW (Train)	90.7	114.7
$[\text{LSP}]_{\text{EFT}}$	88.6	91.5
$[\text{MPII}]_{\text{EFT}}$	68.0	78.9
$[\text{COCO-Part}]_{\text{EFT}}$	59.7	66.9
$[\text{COCO-All}]_{\text{EFT}}$	**58.1**	65.2
$[\text{COCO-Part}]_{\text{EFT}}$ + H36m	**58.6**	45.0
$[\text{COCO-All}]_{\text{EFT}}$ + H36m + MI	**54.2**	43.7
$[\text{COCO-All}]_{\text{EFT}}$ + H36m + MI + 3DPW	**52.2**	43.8
$[\text{COCO-Part}]_{\text{SPIN [8]}}$	69.9	79.2
$[\text{COCO-Part}]_{\text{SMPLify}}$	69.1	77.7

Quantitative Evaluation on 3DPW (PA-MPJPE) and H36M (protocol-2 using frontal view, PA-MPJPE). Reconstruction errors are reported in mm after alignment by a rigid transformation. Top: previous methods. Bottom: training using straight 3D supervision using actual and pseudo 3D annotations. We also compare generating pseudo-ground truth annotations using EFT, SPIN [8], and SMPLify [1]. We report in bold regression results that are better than the previously-reported state-of-the-art on these benchmarks. The COCO dataset with our EFT pseudo-ground truth ($[\text{COCO-All}]_{\text{EFT}}$) is sufficient to beat the previous single image-based state-of-the art (SPIN).

Table 2

Datasets for model training	No post proc.	SMPLify post proc.	EFT post proc.
H36M	146.6	**140.5**	**102.0**
$[\text{LSP}]_{\text{EFT}}$	88.6	**85.5**	**74.7**
$[\text{MPII}]_{\text{EFT}}$	68.0	71.6	**63.2**
$[\text{COCO-Part}]_{\text{EFT}}$	59.7	69.5	**58.1**
$[\text{COCO-All}]_{\text{EFT}}$	58.1	69.1	**56.6**
$[\text{COCO-All}]_{\text{EFT}}$ + H36M + MI	54.2	66.6	**52.9**
$[\text{COCO-All}]_{\text{EFT}}$ + H36M + MI + 3DPW	52.2	67.5	**50.9**
$[\text{COCO-All}]_{\text{EFT}}$ + H36M + MI + 3DPW			49.3 (by Iter. 3)
$[\text{COCO-All}]_{\text{EFT}}$ + H36M + MI + 3DPW			45.2 (by Oracle)

Table 3: Quantitative evaluation on 3DPW by using SMPLify and EFT as post processing, in PA-MPJPE errors (mm). The second column shows the errors without any post processing, and the third and fourth columns show the post-processing results by SMPLify and our EFT method. Bold fonts indicate cases in which post-processing improves the original outputs. The bottom of the table shows the EFT post-processing performance by choosing the best iteration number for this testing data, where "by oracle" means by choosing the best iteration for each test sample.

4 Results

We consider two applications of EFT: creating pseudo-ground-truth 3D annotations for in-the-wild datasets that natively come only with 2D annotations and post-processing the output of an existing 3D pose regressor to improve it.

Implementation details. For the pose regressor Φ, we use the state-of-the-art SPIN network of [8]. For EFT, we optimize Eq. (3) using using Adam [53] with the default PyTorch parameters and a small learning rate of 10^{-6} stopping when the average 2D keypoints re-projection error is less than 2 pixels (usually less than 20 iterations are sufficient). We also found beneficial to modify Eq. 3 to ignore the locations of hips and ankles, which are noisy especially for manual annotations, and use instead a term that matches only the *2D orientation* of the lower legs (see the appendix for details).

Datasets. We use the in-the-wild datasets with **2D pose annotations**: COCO [18], MPII [39], and LSP [40, 41]. We consider the default splits as well a the "COCO-Part" subset that [15] uses for training and that contains only instances for which the full set of 12 keypoint annotations are present (occluded instances often miss keypoints). To this, we add "COCO-All" containing all samples with at least 5 keypoint annotations. We also use datasets with **3D pose annotations**, including H36M [19, 54], MPI-INF-3DHP [20], and Panoptic Studio [21]. Since a multi-view setup is usually required to capture this kind of ground truth, these datasets are collected in laboratory conditions. We use the "moshed" version of H36M and MPI-INF-3DHP [7, 8], and produce SMPL fittings for Panoptic Studio DB using the provided 3D keypoints (see the supp. for details). The 3DPW dataset [51] is captured outdoor and comes with 3D ground truth obtained by using IMUs and cameras by using IMU sensors and cameras.

4.1 EFT for creating 3D pseudo-ground truth

An application of EFT is to generate 3D annotations for existing in-the-wild human datasets. This is important because, compared to data that comes with 3D ground truth but is collected in controlled conditions, in-the-wild datasets more closely reflect the statistics of data found in applications. However, these datasets generally lack 3D ground truth, as the latter requires specialized sensors and setups. Hence, we experiment with the following idea. Given a 2D dataset such as COCO and a pretrained pose regressor, we use EFT to lift the 2D annotations to 3D and use these as 3D ground truth for training novel regressors. We use the notation $[\cdot]_{\text{EFT}}$ to denote 2D datasets lifted to 3D in this manner and consider $[\text{COCO-Part}]_{\text{EFT}}$, $[\text{COCO-All}]_{\text{EFT}}$, $[\text{MPII}]_{\text{EFT}}$, and $[\text{LSP}]_{\text{EFT}}$.

In order to validate the pseudo ground-truth annotations, we use our EFT datasets to train from scratch pose regressors and assess their 3D reconstruction performance on standard benchmarks. We also consider combinations of the EFT datasets with other datasets, such as H36M, that come with 3D ground-truth annotations. In all cases, we use as single training loss the prediction error against 3D annotations (actual or pseudo), with a major simplification compared to approaches that mix, and thus need to balance, 2D and 3D supervision. In particular, we use SPIN [8] as regressor, but we retain only the network *architecture*, switching off their use of SMPLify to learn from 2D annotations.

Results on public benchmarks: 3DPW and H36M. The results are summarized in Table 2. The table (bottom) evaluates the regressor trained from scratch using straight 3D supervision on standard 3D datasets (H36M, MPI-INF-3DHP, PanopticDB), the EFT-lifted datasets $[\text{MPII}]_{\text{EFT}}$, $[\text{LSP}]_{\text{EFT}}$, $[\text{COCO-Part}]_{\text{EFT}}$ and $[\text{COCO-All}]_{\text{EFT}}$, as well as various combinations. Following [7, 8], performance is measured in terms of reconstruction errors (PA-MPJPE) in mm after rigid alignment on two public benchmarks: 3DPW and H3.6M[1]. The models trained with indoor 3D pose datasets (H36M, MPI-INF-3DHP, and PanopticDB) perform poorly on the 3DPW dataset, which is collected outdoor. By comparison, training exclusively on the EFT datasets performs much better. Notably, the model trained *only* $[\text{COCO-All}]_{\text{EFT}}$ outperforms the previous state-of-the art method, SPIN [15], on this benchmark. Combining training data improves performance further, with the second best result (54.2 mm) achieved by the $[\text{COCO-All}]_{\text{EFT}}$ + H36M + MPI-INF-3DHP combination, and the best one (52.2 mm) obtaining by including the 3DPW training data too.

We also compare EFT to SMPLify [1] in two ways. In $[\text{COCO-Part}]_{\text{SMPLify}}$ we use SMPLify to post-process the output of the fully trained SPIN model, and in $[\text{COCO-Part}]_{\text{SPIN}}$ we use the "SMPLified" outputs that SPIN generates during learning [8]. As shown, the models trained on the EFT datasets perform better than the ones trained on the SMPLify-based ones, suggesting that the pseudo ground-truth generated by EFT is better.

As it might be expected, networks trained exclusively on the the EFT-based datasets, which are 'in the wild', are not as good when tested on H36M, which is collected in laboratory conditions. However, the error decreases markedly once one the networks are also trained using the H36M data (H36M, $[\text{COCO-Part}]_{\text{EFT}}$+H36M, and $[\text{COCO-All}]_{\text{EFT}}$ + H36M +MPI-INF-3DHP). This shows the significance of the indoor-outdoor domain gap and highlights the importance of developing in-the-wild 3D datasets, as these are usually closer to applications, thus motivating our approach.

For comparison, Table 2 also reports the performance of state-of-the-art SMPL regressors on these datasets. Our direct competitors are single-image methods such as HMR [7], HoloPose [52] and SPIN [8]. We also include methods that use multiple video frames (hence more information) such as

[1] See the supp. material for the result on the MPI-INF-3DHP benchmark.

HMR-temporal [50] and VIBE [35]. Our best regressors outperform all prior methods, except for models that use temporal information *and* are trained directly on the 3DPW training set. If we only consider methods that take a single image as input (like ours do) training *only* on the $[\text{COCO-All}]_{\text{EFT}}$ dataset is sufficient to beat the state of the art.

4.2 EFT for regression post-processing

EFT can also be added as a drop-in post-processing step to any given pose regressor Φ, improving its fit to 2D annotations and, hopefully, the final 3D reconstructions as well. We compare EFT post-processing against performing the same refinement using a traditional fitting method such as SMPLify, starting from the same regressor for initialization. Since the latter is prone to get stuck in local minima, we use multi-stage fitting, optimizing the global body orientation and camera projection first, and then the joints [2].

Qualitative comparison. For qualitative evaluation, we show 500 randomly-chosen images from the MPII, COCO and LSPet datasets to human annotators in Amazon Mechanical Turk (AMT) and ask them whether they prefer the EFT or the SMPLify reconstruction. Each sample is shown to three different annotators, showing the input image and two views of the 3D reconstructions, from the same viewpoint as the image and from the side. Examples are shown in the left side of Fig. 2. Our method was preferred **61.8%** of the times with a majority of at least 2 votes out of 3, and obtained **59.6%** favorable votes by considering the 1500 votes independently. We found that in many cases the perceptual difference between SMPLify and EFT are subtle, but SMPLify suffers more from bad initialization due to occlusions and challenging body poses. SMPLify also tends to regress to the mean pose too much (e.g., knees tend to be bent).

Quantitative comparison. When 3D ground-truth annotations are available, we can carry out a full quantitative evaluation. In Table 3 we do so by using the 3DPW dataset for in-the-wild scenes. For initialization, we pre-train the same regressor Φ on a number of different datasets. We then use EFT and SMPLify post-processing to fit the same set of 2D keypoint annotations, obtained automatically by means of the OpenPose detector [24]. The key observation is that, while EFT improves the accuracy of *all* tested models, SMPLify, while still improving the 2D fits, results in marginally improved or even worse 3D reconstructions. Notably, via EFT post-processing, our model trained on the $[\text{COCO-All}]_{\text{EFT}}$ data (Section 4.1) has accuracy comparable to the the state-of-the art *video-based regressor* VIBE (56.6 vs 56.5 mm), while utilizing a single image (instead of video) as input. EFT post-processing can also improve our best model trained on $[\text{COCO-All}]_{\text{EFT}}$+H36m+MPI-INF+3DPW, lowering the error to just 50.9 mm and achieving the state-of-the art against all previous approaches (including video-based methods). Finally, we also obtain an *upper bound* of 49.3 mm by optimizing the number of EFT for this dataset (the optimal number is 3); furthermore, if we use an oracle to tell us the best number of iterations for individual samples, the error is reduced to 45.2 mm. This shows that there is space for further substantial gains by finding better stopping conditions.

4.3 Further analysis of EFT

By overfitting to a single sample, EFT could 'break' the generalization capabilities of the regressor, and so the prior it captures. To test whether this is likely to happen, we overfit the regressor to a single sample by using resp. 20 and 100 EFT iterations and then evaluate the accuracy of the overfitted regressor on the entirety of the 3DPW test set, recording the resulting accuracy. We repeat this experiment 500 times for different samples and report the results in Fig. 4. The performance of SPIN [8] and HMR [7] are also shown for comparison. As can be noted, the overfitted regressors still perform very well overall, suggesting that the network *retains* its good properties despite EFT. In particular, the performance is at least as good as the HMR baseline [7], and occasionally the overall performance improves after overfitting a single sample. The effect is different for different samples — via inspection, we found that samples that are more likely to 'disrupt' the regressor contain significant occlusions or annotation errors. Note that the fine-tuned network is discarded after applying EFT on a single example — this analysis is only meant to illustrate the effect of fine-tuning, but has no direct implications on the effectiveness of EFT.

4.4 Qualitative Evaluation on Internet Videos

[2] We use the SMPLify code provided by the SPIN [8] with minor improvements. See the supp. material.

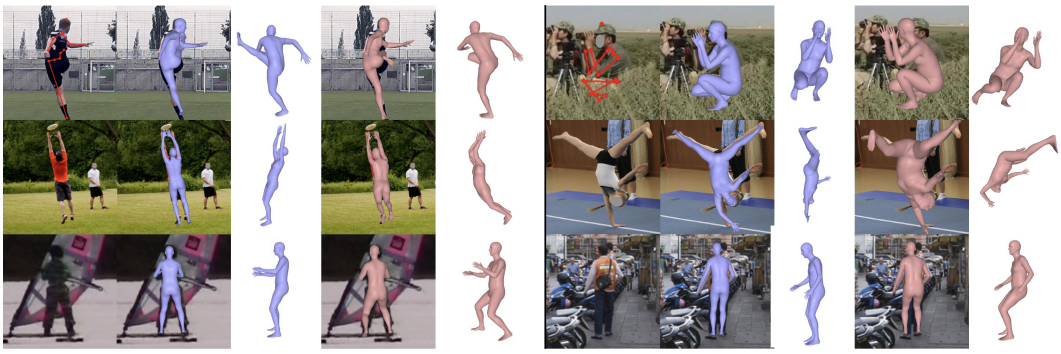

Figure 2: Qualitative comparison between our EFT-based pseudo-annotations (blue) and SMPLify (pink) used for the AMT study. Traditional fitting methods such as SMPLify considers only 2D keypoints as input, producing inaccurate 3D poses despite aligning the 2D joints correctly. SMPLify is also prone to local minima, where data term and prior term are in conflict, and its output tends be close to the mean pose (*e.g.* note the bent knees in the last row). EFT produces more accurate 3D outputs given the same input.

Figure 3: 3D pose estimation results a the model trained using only our our EFT databases on challenging in-the-wild video sequences. Bounding boxes are provided by an off-the-shelf detector [55]

We demonstrate the performance of our 3D pose regressor models trained using our EFT datasets on various challenging real-wold Internet videos, containing cropping, blur, fast motion, multiple people, and other challenging effects. Data of this complexity was rarely considered in prior work. Example results are shown in Fig. 3 and also in the supp. videos.

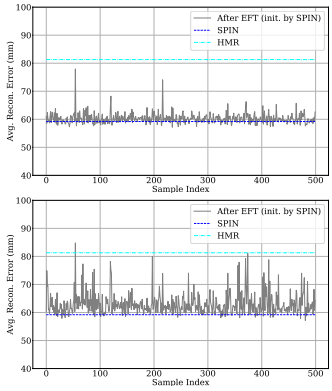

Figure 4: Testing error on 3DPW (PA-MPJPE in mm) after overfitting 500 different samples via EFT using 20 (top) and 100 (bottom) iterations.

5 Discussion

We introduced Exemplar Fine-Tuning (EFT), a method to fit a parametric 3D human body model to 2D keypoint annotations. Leveraging a trained 3D pose regressor as pose prior conditional on RGB inputs, EFT produces more plausible and accurate fitting outputs than existing methods. EFT can be used in post-processing to improve the output of existing 3D pose regressors. It can also be used to generate high-quality 3D pseudo-ground-truth annotations for datasets collected in the wild. The quality of these labels is sufficient to supervise state-of-the art 3D pose regressors. We expect these 'EFT datasets' to be of particular interest to the research community because they greatly simplify training 3D pose regressors, avoiding complicated preprocessing or training techniques, as well as the need to mix 2D and 3D annotations. We will release the 'EFT datasets' to the community, allowing their use in many other tasks, including dense keypoint detection [44], depth estimation [56], or the recognition of human-object interactions in the wild.

References

[1] 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] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *CVPR*, 2017.

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

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

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

[6] Hsiao-Yu Fish Tung, Hsiao-Wei Tung, Ersin Yumer, and Katerina Fragkiadaki. Self-supervised learning of motion capture. In *NIPS*, 2017.

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

[8] 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.

[9] V. Tan, I. Budvytis, and R. Cipolla. Indirect deep structured learning for 3D human body shape and pose prediction. In *BMVC*, 2017.

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

[11] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*, 2017.

[12] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3D human pose estimation. In *CVPR*, 2018.

[13] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter V. Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *3DV*, 2018.

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

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

[16] Yu Cheng, Bo Yang, Bo Wang, Wending Yan, and Robby T. Tan. Occlusion-aware networks for 3d human pose estimation in video. In *ICCV*, 2019.

[17] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *CVPR*, 2019.

[18] 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.

[19] 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.

[20] 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.

[21] Hanbyul Joo, Tomas Simon, Xulong Li, Hao Liu, Lei Tan, Lin Gui, Sean Banerjee, Timothy Godisart, Bart Nabbe, Iain Matthews, et al. Panoptic studio: A massively multiview system for social interaction capture. *TPAMI*, 2017.

[22] Yu Sun, Yun Ye, Wu Liu, Wenpeng Gao, Yili Fu, and Tao Mei. Human mesh recovery from monocular images via a skeleton-disentangled representation. In *ICCV*, 2019.

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

[24] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 3D pose estimation using Part Affinity Fields. In *CVPR*, 2018.

[25] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

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

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

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

[29] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. 2005.

[30] P. Guan, A. Weiss, A. O. Balan, and M. J. Black. Estimating human shape and pose from a single image. In *ICCV*, 2009.

[31] Leonid Sigal, Alexandru Balan, and Michael J. Black. Combined discriminative and generative articulated pose and non-rigid shape estimation. In *NeulIPS*, 2008.

[32] 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.

[33] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *CVPR*, 2012.

[34] Yuanlu Xu, Song-Chun Zhu, and Tony Tung. Denserac: Joint 3d pose and shape estimation by dense render-and-compare. In *ICCV*, 2019.

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

[36] Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*, page 18. ACM, 2015.

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

[38] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3d human pose estimation in the wild: a weakly-supervised approach. In *ICCV*, 2017.

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

[40] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *BMVC*, 2010.

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

[42] Weiyu Zhang, Menglong Zhu, and Konstantinos G Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013.

[43] M. Andriluka, U. Iqbal, E. Ensafutdinov, L. Pishchulin, A. Milan, J. Gall, and Schiele B. PoseTrack: A benchmark for human pose estimation and tracking. In *CVPR*, 2018.

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

[45] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *IJCV*, 2010.

[46] 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.

[47] CMU graphics lab motion capture database.

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

[49] 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.

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

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

[52] Riza Alp Guler and Iasonas Kokkinos. Holopose: Holistic 3d human reconstruction in-the-wild. In *CVPR*, 2019.

[53] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015.

[54] Catalin Ionescu, Fuxin Li, and Cristian Sminchisescu. Latent structured models for human pose estimation. In *ICCV*, 2011.

[55] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[56] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, 2019.

[57] Song-Hai Zhang, Ruilong Li, Xin Dong, Paul L Rosin, Zixi Cai, Han Xi, Dingcheng Yang, Hao-Zhi Huang, and Shi-Min Hu. Pose2seg: Detection free human instance segmentation. In *CVPR*, 2019.

[58] Yu Rong, Ziwei Liu, Cheng Li, Kaidi Cao, and Chen Change Loy. Delving deep into hybrid annotations for 3d human recovery in the wild. In *ICCV*, 2019.

[59] Carl Doersch and Andrew Zisserman. Sim2real transfer learning for 3d human pose estimation: motion to the rescue. In *Neurips*, 2019.

[60] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *CVPE*, 2019.

Supplementary Material

In this supplementary material, we address several additional components to support our main paper, including EFT implementation details, newly proposed "extreme crop" augmentation, more experiments on public benchmarks, computation time, and more qualitative comparisons between SMPLify [1] and our Exemplar-Fine Tuning (EFT).

A Implementation Details

Exemplar Fine-Tuning: We use the publicly available code and pre-trained model of SPIN [8] as the 3D pose regressor that provides the learned 3d pose prior for EFT optimization. As a modification, we change the perspective projection model used in SPIN to the weak-perspective projection as in HMR [7]. A minor fine-tuning process from the pre-trained SPIN model is needed to make the model adjusted to this camera model change, where we use the same datasets and SMPL fittings from SPIN. This minor modification is only intended to make the camera projection simpler, and does not noticeably affect the performance of original SPIN model.

For EFT process, we put a single target sample, a cropped RGB image around the target person and corresponding 2D keypoint annotations, in a batch, and run EFT iterations via Adam [53] optimizer. We turn off batch normalization and dropout. We use 20 iterations as default in building our EFT dataset, but stop the iteration earlier if the average 2D keypoint error is less than 2 pixels. More iterations are sometimes used to test the bad initialization scenarios (e.g., OCHuman dataset [57] in our supp. video). We use Adam [53] optimizer with the default PyTorch parameters and a small learning rate of 10^{-6}.

Although the 2D keypoints are annotated by humans in public 2D keypoint datasets, they are still noisy due to inaccuracies in the points clicked by the annotators. This noise can adversely affect the quality of 3D human model fitting process. For example, if the distance between two joints is shorter than it should, this may cause the predictor to tilt bones excessively to compensate. We found that 2D keypoints on the hips and ankles are often less precisely localized. Thus, we ignored the hip and foot annotation joints in the calculation for the purpose of fine-tuning. Instead, we add to the loss a penalty to match the *orientation* of the lower leg, encouraging the reconstruction of the *orientation* of the vectors connecting the knee to the ankle.

After producing the pseudo ground-truth 3D pose by EFT, we filter out unreliable fittings by checking the maximum value of the SMPL shape parameters (reject if lager than 5), and also 2D keypoint loss with a threshold (0.01).

Training A 3D Pose Regressor: As mentioned, we train the SPIN network [8] from scratch to demonstrate the high pseudo ground-truth quality of ours EFT dataset. To train a 3D pose regressor, we use the same hyper parameters of SPIN [8] with 256 batch size. We trained the model about 200K iterations, where we found they often converge. Two GPUs (Quadro GP100s) are used to train a model, and the training takes about 2-3 days.

We use the same data augmentation (rotation, flip, noise) with SPIN, and importantly we newly add the "extreme body crop" augmentation on the final version to handle the extremely cropped scenes (e.g., only upper body is visible), which is commonly observable in real scenarios. See the next section for details.

B Augmentation by Extreme Cropping

A shortcoming of previous 3D pose estimation methods is that they assume that most of the body is visible in the input image [34, 52, 7, 8]. However, humans captured in real-world videos are often cropped or occluded, so that only the upper body or even just the face may be visible (see our supp. videos). Occlusions dramatically increase the ambiguity of pose reconstruction, as in this case not just depth, but the whole position of several keypoints is not observable at all. Hence, the quality of the reconstructions depends even more strongly on the quality of the underlying pose prior. Here, we wish to train a model that can handle such difficult cases. We propose to do so by augmenting the training data with extreme cropping. Since we already have full 3D annotations, doing so is straightforward — we only need to randomly crop training samples. We do so by first cropping either

DBs for model training	Without crop aug.	With crop aug.
SPIN [8]	131.4	—
$[\text{COCO-Part}]_{\text{EFT}}$	102.3	70.2
$[\text{COCO-All}]_{\text{EFT}}$	86.1	71.0
$[\text{COCO-All}]_{\text{EFT}}$ + H36M + MPI-INF-3DHP	92.0	68.6

Table 4: Effect of crop-augmentation on 3DPW (PA-MPJPE in mm). Models are tested by using the upper body only images as inputs. The second column show the performances without crop augmentation (the same errors as in the Table 2 of our paper), and the third column shows the output after applying crop augmentations.

the upper body up to the hips, or the face and shoulders up to the elbows, and then further crop the result using a random bounding box of size equal to 80%-120% of that of the first crop. While the input image is cropped, we retain the full 2D/3D body joint supervision to allow the network to learn to reconstruct the occluded body parts in a plausible manner. We apply crop augmentation for our fully trained models (e.g., the model trained with $[\text{COCO-All}]_{\text{EFT}}$ + H36M + MPI-INF-3DHP) by running about 50K more iterations.

Evaluation : Since none of the existing datasets with 3D annotations contain significant occlusions, we asses robustness to occlusions by cropping upper bodies from 3DPW and feeding them to the network as input. We use the same metric (PA-MPJPE) as in the original benchmark test, except that we adjust the bounding box size to have only the upper body only (2D keypoints from torso and limbs). The result is shown in Table 4. While both original SPIN and our network trained with $[\text{COCO-Part}]_{\text{EFT}}$ without crop augmentation work poorly, we found our model trained with $[\text{COCO-All}]_{\text{EFT}}$ shows better performance (86.1 mm) even without crop augmentation. This is because $[\text{COCO-All}]_{\text{EFT}}$ already includes many such samples with severe occlusions. Note that $[\text{COCO-All}]_{\text{EFT}}$ includes all samples with 6 or more valid 2D keypoint annotations. Our models trained with crop augmentation show much better performance. Remarkably, the performance of our best model by observing only the upper body images shows better performance (68.2 mm) than the HMR model by observing the *whole* body (81.3 mm).

The performance of these models trained with crop augmentation on the original benchmarks are shown in Table 6 bottom parts, denoted as "(Crop aug.)". Although this crop augmentation slightly increases the errors compared to the original models without this augmentation, we found that it is effective in real-world video scenarios.

C Further Evaluation on Standard Benchmarks

The full evaluation results on the standard benchmarks including MPI-INF-3DHP [20]) are shown in Table 6. All results are measured in terms of reconstruction errors (PA-MPJPE) in mm after rigid alignment, following [7, 8]. Here, we highlight several noticeable results.

Evaluation on MPI-INF-3DHP [20] The model trained with our EFT dataset is also competitive in MPI-INF-3DHP dataset. The models trained by *only* , $[\text{MPII}]_{\text{EFT}}$, $[\text{COCO-Part}]_{\text{EFT}}$ $[\text{COCO-All}]_{\text{EFT}}$ outperform the performance of HMR [7]. Combining 3D datasets with our EFT dataset improves the performance further.

Using 2D Only Annotations Without Pseudo-GT Instead of producing the pseudo-GTs for 2D datasets via EFT or SMPLify method [1, 8], one may try to train the 3D pose regressor by just using 2D annotations without including other 3D dataset. The results are shown as $[\text{COCO-Part}]_{\text{2D}}$ and $[\text{COCO-All}]_{\text{2D}}$ in Table 6. As expected, the model cannot estimate accurate 3D pose, showing very poor performance. However, interestingly, we found the 2D projection of the estimated 3D keypoints is still quite accurately aligned to the target individual's 2D joints.

Another alternative baseline approach is combining the datasets with 2D annotations only (COCO and MPII) with the 3D datasets with full 3D supervisions (H36M and MPI-INF-3DHP), as similarly tried in previous methods [7, 58]. We also perform the similar test here. Different from HMR [7], we

do not use adversarial loss, but still the result shows competitive performance in both indoor datasets (H36M, MPI-INF-3DHP) and outdoor dataset (3DPW), showing better performance than HMR. This result demonstrates that 2D only dataset can be effectively combined with 3D dataset, as also shown in previous work [38, 58]. However, still the performance is significantly worse than the models trained with our EFT datasets with pseudo ground-truths (72.6 mm vs. 54.2 mm in 3DPW dataset).

H36M Protocol-1 We also report the results via H36m protocol 1. In this evaluation, we use all available camera views in H36M testing set. The overall errors are slightly higher than protocol-2 (frontal camera only), but the general tendency is similar to the results of H36M protocol-2.

D Computation Time EFT

We compare the computation time between our EFT and SMPLify processes. The computation times are computed during the processing by a single GeForce RTX 2080 GPU.

EFT. A single EFT iteration takes about 0.04 sec., and a whole EFT process for a sample data (including reloading the pre-trained network model) takes about **0.82 sec.** with 20 iterations.

SMPLify. We use the default iteration number used in SPIN [8] with 100 iterations for optimizing camera parameters and global orientation only, and 100 iterations for whole body pose optimization after fixing the camera parameters. Camera optimization takes about 0.01 sec. per iteration, and, thus, it takes 1 sec per sample (with 100 iterations). Optimizing body pose takes about 0.02 sec. per iteration, and it takes 2 sec. with 100 iterations. Thus whole SMPLify process takes about **3 sec.** per sample.

E Sampling Ratios across Datasets During Training

Following the approach of SPIN [8], we use a fixed sampling ratio across datasets during training, instead of naively mixing them. This is helpful to avoid the problem that a dataset with much larger samples (e.g., H36M and Panoptic Studio) overwhelm the model. The ratio used in our experiments are shown in Table 5. $[\mathrm{X}]_{\mathrm{EFT}}$ denotes our EFT dataset, where, if multiple EFT datasets are used (e.g., COCO and MPII), they are naively merged.

Dataset Combinations	$[\mathrm{X}]_{\mathrm{EFT}}$	**H36M**	**MPI-INF**	**PanopticDB**	**3DPW**
$[\mathrm{X}]_{\mathrm{EFT}}$	100%	0	0	0	0
$[\mathrm{X}]_{\mathrm{EFT}}$ + H36m	60%	40%	0	0	0
$[\mathrm{X}]_{\mathrm{EFT}}$ + Pan3D	40%	0	0	60%	0
$[\mathrm{X}]_{\mathrm{EFT}}$ + H36m + PanDB	40%	40%	0	20%	0
$[\mathrm{X}]_{\mathrm{EFT}}$ + H36m + MPI-INF	30%	50%	20%	0	0
$[\mathrm{X}]_{\mathrm{EFT}}$ + H36m + MPI-INF + 3DPW	30%	40%	10%	0	20%

Table 5: Training data sampling ratios across datasets.

F Analysis on the Failures of SMPLify Fittings in SPIN

We found the erroneous cases in the publicly available SMPL fitting outputs from SPIN [8] that is related to its SMPLify [1] implementation. In the SPIN, both 2D keypoint annotations and OpenPose [24] estimations are used for SMPLify, assuming that the OpenPose estimation has better localization accuracy if it is applied to the training data. However, we found many cases that the OpenPose output from different individual is mistakenly associated to the target person, as shown in Fig 5. Due to the reason, we exclude the OpenPose estimation and only use the GT annotations for our SMPLify process.

G SMPL Fitting for Panoptic Studio Dataset

We apply our EFT to produce SMPL fitting on Panoptic Studio dataset [21]. Different from in-the-wild datasets, Panoptic Studio dataset has ground truth 3D keypoint annotations. However, still it is

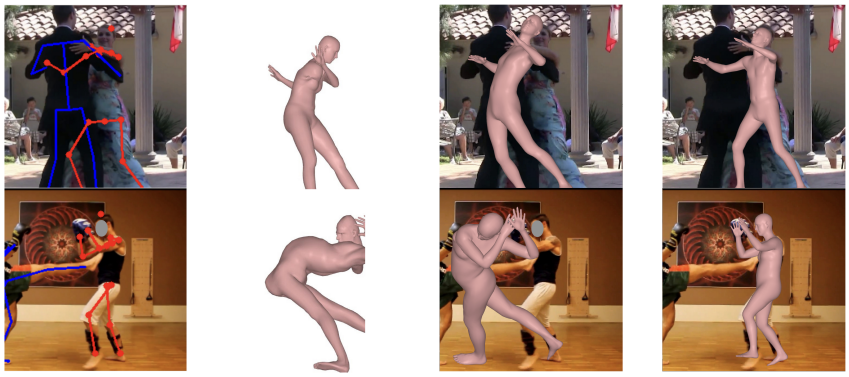

Figure 5: In SMPLify implementation of SPIN [8], sometimes the OpenPose estimations (blue 2D keypoints) from different individuals are incorrectly associated to the target person (with red GT 2d keypoints). Thus, we completely ignore OpenPose estimation during our SMPLify process. (1st column) input images with GT annotaiton (red) and OpenPose estimation (blue); (2nd column) Fitting output produced from SPIN [8] (the publicly available data); (3rd column) our SMPLify output by using both GT and incorrectly associated OpenPose estimation; (4th column) our SMPLify output by using GT 2D keypoints without using OpenPose output.

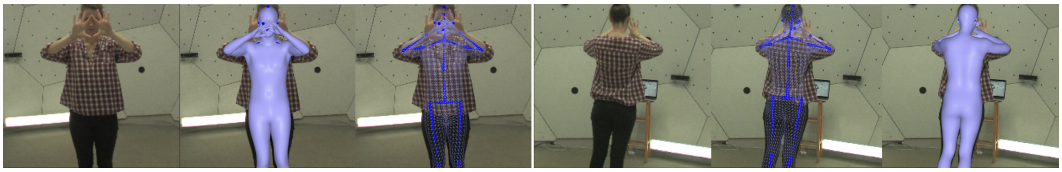

Figure 6: Examples of 3D model fitting outputs on Panoptic Studio data [17] with 3D GT skeletons. We made a simple modification of our original EFT by including 3D keypoint loss.

not sufficient to fully constraint the pose parameters (3 DOF angles for each joint). Our EFT method leveraging the learned 3D pose prior can be effectively used in this case. As a simple modification to our original EFT loss function (Eq. 3 in our main paper), we add the the 3D keypoint loss term L_{3D} to use the available 3D keypoint annotations:

$$\mathbf{J}(\hat{\mathbf{j}}, \mathbf{I}) = M(\Phi_{\mathbf{w}^+}(\mathbf{I})) \;\; \text{where} \;\; \mathbf{w}^+ = \underset{\mathbf{w} \in \mathcal{N}(\mathbf{w}^*)}{\operatorname{argmin}} \; L_{\text{2D}}\left(\pi\left(M(\Phi_{\mathbf{w}}(\mathbf{I}))\right), \hat{\mathbf{j}}\right) + L_{\text{3D}}\left(M(\Phi_{\mathbf{w}}(\mathbf{I})), \hat{\mathbf{j}}\right) + \lambda \|\boldsymbol{\beta}\|_2^2. \tag{4}$$

Some example fitting outputs are shown in Fig 6.

H Samples Causing Significant Model Changes during EFT

As can be seen in *Fig.4 of our main paper*, overfitting the network to individual samples during EFT usually has a small effect on the *overall* regression performance, suggesting that the network *retains* its good properties despite fine-tuning exemplars. The effect is different for different samples, and we show the examples with a strong effect in Fig. 7 that contain significant occlusions or annotation errors.

I More Qualitative Comparisons Between EFT vs. SMPLIfy

As addressed in our main paper, our EFT method was preferred **61.8%** of the times with a majority of at least 2 votes out of 3 in Amazon Mechanical Turk (AMT) study. Here, we visualize the examples

Figure 7: (Left) Example samples that cause significant changes in the 3DPW testing error of network after EFT. The left two examples have annotations on occluded body parts, and the rightest example has incorrect annotations (left-right swap).

with 0 vote and 3 votes, as shown in Fig 8 and 9. Note that in actual AMT study, we show the meshes in white backgrounds to reduce the bias cased by 2D localization quality.

When annotators prefer SMPLify There are 47 / 500 samples where SMPLify outputs are favored by all three annotators, and examples are shown in Fig. 8.Surprisingly the difference is quite minimal, and no obvious reason is found.

When annotators prefer EFT There were 132 / 500 samples where our EFT outputs are favored by all three annotators. Examples are shown in Fig. 9. In most cases, EFT produces more convincing 3D poses leveraging the learned pose prior conditional on the target raw image, while SMPLify tends to produce the outputs similar to its prior poses (e.g., knees tend to be bent).

Previous method	**3DPW ↓**	**H36M (P1) ↓**	**H36M (P2) ↓**	**MPI-INF-3DHP ↓**
HMR [7]	81.3	58.1	56.8	89.8
DenseRaC [34]	—	—	48.0	—
DSD [22]	75.0	—	44.3	—
HoloPose [52]	—	—	50.6	—
HoloPose (w/ post-proc.) [52]	—	—	46.5	—
SPIN [8]	59.2	—	41.1	67.5
HMR (temporal) [50]	72.6	—	56.9	—
Sim2Real (temporal) [59]	74.7	—	—	—
Arnab et al. (temporal) [60]	72.2	—	54.3	—
DSD (temporal) [22]	69.5	—	42.4	—
VIBE (temporal) [35]	56.5	44.2	41.5	63.4
VIBE (temporal) (w/ 3DPW train) [35]	51.9	44.1	41.4	64.6
Straight 3D supervision and pseudo-ground truth from EFT				
H36M	146.6	57.8	54.9	154.9
PanopticDB (PanDB)	115.4	109.1	107.9	129.1
MPI-INF-3DHP (MI)	97.3	108.0	106.2	113.0
3DPW (Train)	90.7	117.6	114.7	124.3
$[\text{MPII}]_{\text{EFT}}$	68.0	80.8	78.9	89.6
$[\text{LSP}]_{\text{EFT}}$	88.6	94.7	91.5	99.4
$[\text{COCO-Part}]_{\text{EFT}}$	59.7	70.3	66.9	83.6
$[\text{COCO-All}]_{\text{EFT}}$	**58.1**	68.7	65.3	81.5
$[\text{COCO-Part}]_{\text{EFT}}$ + H36m	**58.6**	47.8	45.0	77.5
$[\text{COCO-All}]_{\text{EFT}}$ + H36m	**56.0**	47.8	45.2	77.2
$[\text{COCO-Part}]_{\text{EFT}}$ + Pan3D	**57.0**	68.7	65.2	82.0
$[\text{COCO-All}]_{\text{EFT}}$ + Pan3D	**56.7**	67.8	66.1	80.8
$[\text{COCO-Part}]_{\text{EFT}}$ + H36m + PanDB	**55.4**	48.9	45.6	76.5
$[\text{COCO-All}]_{\text{EFT}}$ + H36m + PanDB	**55.1**	48.4	46.4	76.8
$[\text{COCO-Part}]_{\text{EFT}}$ + H36m + MI	**55.4**	49.7	46.7	68.8
$[\text{COCO-All}]_{\text{EFT}}$ + H36m + MI	**54.2**	46.4	43.7	68.0
$[\text{COCO-All}]_{\text{EFT}}$ + H36m + MI + 3DPW	**52.2**	46.6	43.8	67.5
Alternative Approaches				
$[\text{COCO-Part}]_{\text{SPIN [8]}}$	69.9	89.3	73.2	94.1
$[\text{COCO-Part}]_{\text{SMPLify}}$	69.1	87.1	77.7	93.3
$[\text{COCO-Part}]_{\text{2D}}$	228.1	236.4	213.7	218.8
$[\text{COCO-All}]_{\text{2D}}$	192.1	198.4	181.1	195.3
$[\text{COCO-Part}]_{\text{2D}}$ + H36m + MI	72.6	60.7	56.9	80.9
with Crop Augmentation				
$[\text{COCO-Part}]_{\text{EFT}}$ + (Crop aug.)*	61.0	72.0	68.3	85.0
$[\text{COCO-All}]_{\text{EFT}}$ + (Crop aug.)*	60.0	70.0	66.4	82.4
$[\text{COCO-All}]_{\text{EFT}}$ + H36M + MI + (Crop aug.)*	**54.2**	48.1	45.2	69.5

Table 6: Quantitative Evaluation on 3DPW, H36M protocol-1 (by using all views), H36M protocol-2 (by using frontal views), and MPI-INF-3DHP dataset. Reconstruction errors are computed by PA-MPJPE and reported in mm after alignment by a rigid transformation.

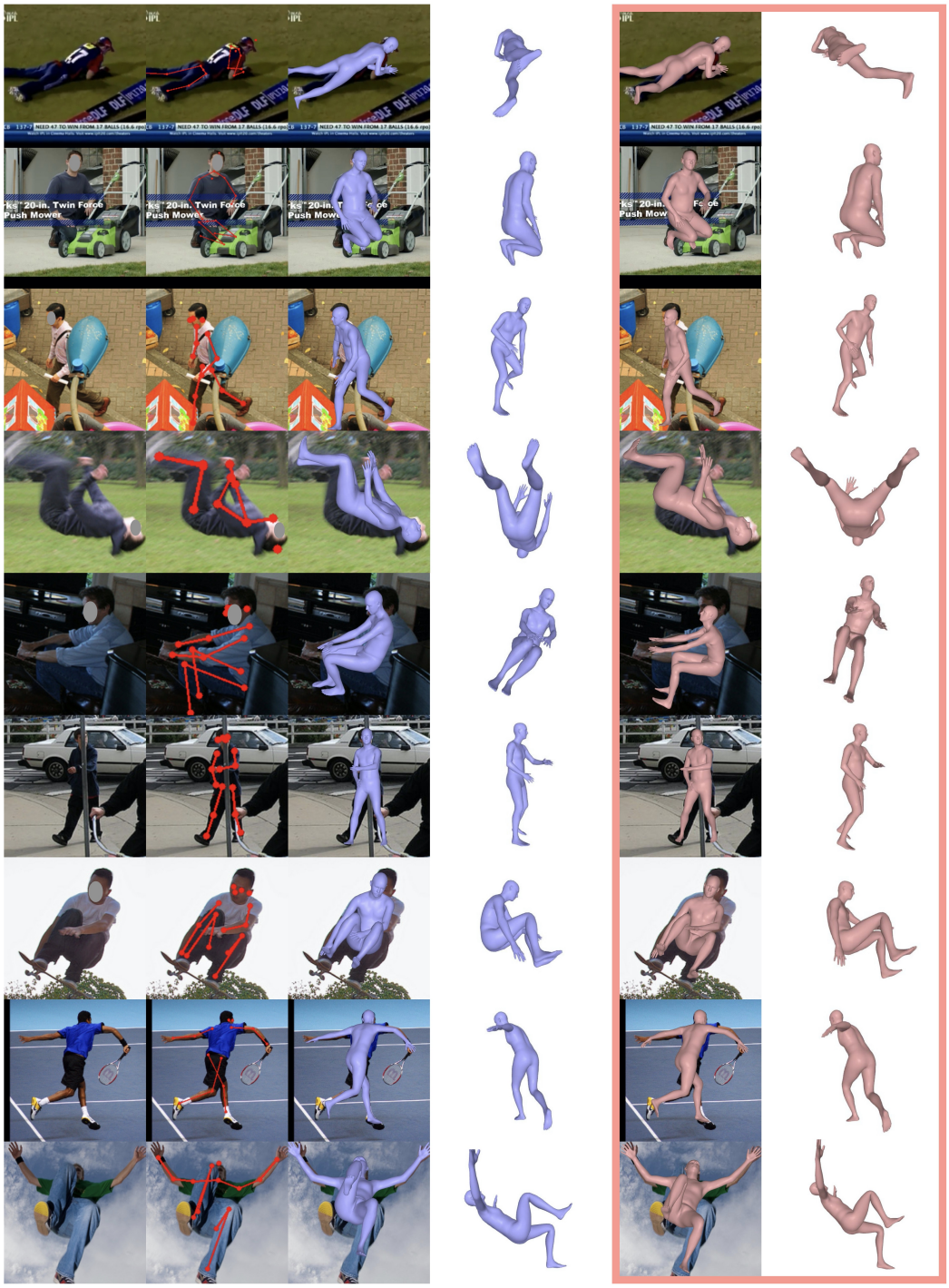

Figure 8: The samples where SMPLify outputs are favored by all three annotators. The blue meshes are the results by EFT and the pink meshes are results by SMPLify.

Figure 9: The samples where our EFT outputs are favored by all three annotators. The blue meshes are the results by EFT and the pink meshes are results by SMPLify.