ReFit: Recurrent Fitting Network for 3D Human Recovery

Yufu Wang Kostas Daniilidis
University of Pennsylvania

Abstract

We present Recurrent Fitting (ReFit), a neural network architecture for single-image, parametric 3D human reconstruction. ReFit learns a feedback-update loop that mirrors the strategy of solving an inverse problem through optimization. At each iterative step, it reprojects keypoints from the human model to feature maps to query feedback, and uses a recurrent-based updater to adjust the model to fit the image better. Because ReFit encodes strong knowledge of the inverse problem, it is faster to train than previous regression models. At the same time, ReFit improves state-of-the-art performance on standard benchmarks. Moreover, ReFit applies to other optimization settings, such as multi-view fitting and single-view shape fitting. Project website: https://yufu-wang.github.io/refit_humans/

1. Introduction

Single-view 3D human reconstruction has transformed the way we analyze and create virtual content. Progress has primarily been driven by the use of parametric human models [43], powerful neural networks [30], and high-quality annotated data [61]. However, the best systems to date still struggle with various difficulties, including occlusion, uncommon poses, and diverse shape variations.

The traditional approach fits a parametric human model to an image using handcrafted objectives and energy minimization techniques [9]. But this optimization process is often riddled with local minima and corner cases. Because of such drawbacks, recent regression methods train a neural network to predict the parameters directly [30, 35]. Training such networks robustly requires a large amount of 3D annotated data that is hard to collect outside a lab setting. One line of research is to design better neural network architectures that can learn efficiently and generalize well to diverse in-the-wild cases.

In this paper, we propose Recurrent Fitting (ReFit), an architecture for 3D human reconstruction. ReFit mimics the structure of model fitting, reducing the regression problem to a learning-to-optimize problem. This allows ReFit to learn faster than other architectures and improves state-of-the-art accuracy.

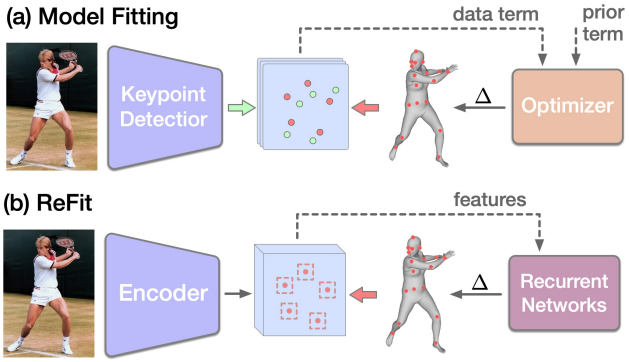

Figure 1. **Overview**. ReFit mimics the strategy of model fitting by constructing a trainable feedback-update loop that adjusts the human model for a more accurate reconstruction.

ReFit has three main steps. (1) A backbone network extracts pixel-aligned image features. (2) A feedback module queries feedback features from keypoint reprojection. (3) A recurrent update module uses a set of disentangled GRUs to update the body. Feedback and updates are repeated until the body mesh is well-aligned with the image (Fig. 1).

The design of ReFit mirrors traditional model fitting. In model fitting, the objective function computes the L2 distance between the reprojected and the detected keypoints [9]. In ReFit, each reprojected keypoint queries a window on a feature map. A window encodes first-order information, such as 2D flow, which is the derivative of the L2 objective. The queried features are fed to the update module to compute updates for the human model.

Another key element of ReFit is the disentangled and recurrent update streams. Iterative regression methods commonly map a feature vector to parameter updates with an MLP [30, 67]. Therefore, the update rules for the parameters are entangled until the last linear layer. In contrast, parameters in optimization follow update rules based on automatic differentiation, and the rules can be very different. As an example, in human pose, an alignment error in the right hand will produce a gradient for the right arm and shoulder but not for the left side. This notion of a disentangled update is difficult to achieve within a global feature vector. When learning

the update function, we hypothesize that some degree of disentanglement is beneficial.

To this end, ReFit's update module uses one GRU per parameter group to disentangle the parametric update. Each GRU learns the prior and the update rule for one parameter group, e.g., the rotation of a single joint. GRUs utilize memory from the previous iterations, analogous to a first-order method with momentum [16]. At each iteration, the GRUs can take different step sizes. In the example of right-hand misalignment, the GRUs for the right side of the body can output adjustments, while the GRUs for the left side can return little or no updates. This disentanglement leads to better performance. The disentangled GRUs can be implemented as batch matrix operations in PyTorch, allowing for efficient training and inference.

Moreover, ReFit acts as a learned optimizer that applies to other settings of model fitting. Using ReFit as a learned optimizer, we propose a multi-view fitting procedure that significantly outperforms regression on separate views. We also demonstrate using ReFit to register a pre-scanned shape to images of the same subject, a helpful setting for outdoor motion capture.

The ReFit architecture is substantially different from prior works. It uses a recurrent-based updater instead of a coarse-to-fine pyramid that limits the number of updates [67]. ReFit applies reprojection to query feedback instead of using a static global feature vector [65]. Compared to learning-to-optimize approaches [16,55], ReFit does not detect keypoints but instead uses learned features and is end-to-end trainable.

ReFit improves state-of-the-art results on standard benchmarks [27, 61]. The learning-to-optimize design accelerates training: ReFit converges with only 50K training iterations. We conduct extensive ablation studies to analyze the core components and applications, and contribute insights for future works.

2. Related Work

Human Mesh Model Fitting. Reconstructing human pose and shape from images has a long history [5, 8, 10, 25]. The reconstruction often is formulated as an energy minimization problem by fitting a parametric model with optimization in various settings: single view [9, 19, 47, 51, 54], multiple views [1, 18, 26], video [6, 52], and other sensing inputs [61,62]. These optimization procedures involve multiple stages and sophisticated designs to avoid local minima.

The optimization objective commonly consists of a data term and a prior term. The data term measures the deviation between the estimation and the detected features, while the prior term imposes constraints on the pose and shape space. In practice, optimization encounters many difficulties, including noisy keypoint detection [11, 57], complicated priors [51, 53, 59], and the trade-off between the two terms. In the single-view setting where such difficulties compound, recent research shifts to two directions: learning to regress and learning to optimize.

Human Mesh Regression. The recognition power of deep neural network [22], paired with the representation capability of parametric human models [43, 64], has fueled the recent progress in single-view human mesh regression [14, 15, 28, 30, 31, 33–35, 39, 49, 63].

Iterative refinement is an essential strategy used by many regression works [30, 48, 65]. Carreira et al. [12] motivate it from a neural science standpoint, which states that the human brain relies on feedback signals for various visual localization tasks. The de facto implementation introduced by HMR [30] is to concatenate the prediction with the image feature vector to make new predictions recurrently. This alone is not an effective strategy, as the network has to learn error feedback solely in the latent space.

PyMAF [66, 67] proposes a refinement strategy based on a coarse-to-fine feature pyramid. The parametric mesh predicted from the coarse level is reprojected to feature maps at the finer levels to gather spatial features for refinement updates. This strategy lifts the error feedback to the image space, but the pyramid limits the number of update steps.

ReFit reprojects mesh keypoints to the feature maps at a single resolution but constructs a recurrent loop with GRUs that is not limited to the number of pyramid levels. Moreover, ReFit's feedback and update modules have novel designs that further boost the accuracy.

Learning to Optimize. Many vision problems are inverse problems traditionally solved by optimization. Learning to optimize aims to train a neural network to propose a descend direction that replaces or supplements the optimizer update [2]. This paradigm inspires network designs that mimic optimizers in various vision tasks [3, 20, 41, 44, 58].

For the human model, LGD [55] and LFMM [16] improve traditional model fitting by using a neural network to predict residues for the optimizer updates. This approach is not end-to-end differentiable because the optimizer uses keypoint detection to compute the gradients. Neural Descent [65] replaces the optimizer with an LSTM-based updater [24] and reduces the detections into a feature vector. However, reducing spatial representation into a feature vector prevents it from having direct error feedback in the image space. ReFit is an evolution of these methods. It uses gated recurrent units [13, 17] as updaters but utilizes reprojection on the learned feature maps as the feedback to infer an update.

ReFit can be regarded as a learned optimizer. Beside single-view inference, we demonstrate that ReFit can be a drop-in replacement for traditional optimizers in multi-view model fitting. We also apply ReFit to register a pre-fitted shape to images of the same subject.

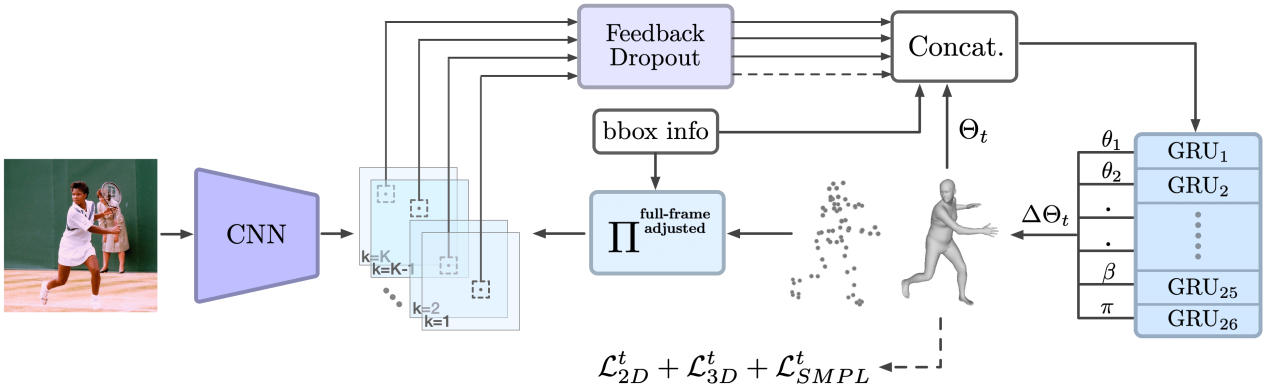

Figure 2. **The ReFit Network.** ReFit extracts one feature map per keypoint with a backbone network (Sec. 3.1). It then reprojects keypoints from the 3D human mesh to the corresponding feature maps using the full-frame adjusted camera model (Sec. 3.2). Feedback is dropped randomly during training, and concatenated with the current estimate Θ_t and the bounding box info to form the final feature vector. The final feature is sent to N parallel GRUs to predict updates for the N parameters (Sec. 3.3). The updated mesh is again reprojected to the feature maps to repeat the feedback-update loop until good reconstruction is achieved.

3. Recurrent Fitting

Given an image of a person, our goal is to predict the parameters of the SMPL human model. The ReFit network extracts image features (Sec. 3.1), compares the human mesh with the pixel-aligned features (Sec. 3.2), and iteratively updates the SMPL parameters (Sec. 3.3). An overview of the method is given in Figure 2.

Preliminaries. The SMPL model [43] is parametrized with $\Theta = \{\theta, \beta, \pi\}$, where $\theta \in \mathbb{R}^{24\times3}$ are the relative rotations of the 24 body joints, $\beta \in \mathbb{R}^{10}$ is the shape parameter, and $\pi \in \mathbb{R}^3$ is the root translation w.r.t the camera. Given Θ, SMPL outputs the 3D mesh $\mathcal{M}(\Theta) \in \mathbb{R}^{6890\times3}$.

3.1. Feature Extraction

We use High-Resolution Net (HRNet) [57] as the feature extractor. It produces spatially precise feature maps that are beneficial for the feedback step. Given an image $I \in \mathbb{R}^{H\times W\times3}$, it outputs feature maps $F \in \mathbb{R}^{H/4\times W/4\times K}$, where K is the number of channels. We set K to equal the number of keypoints.

In addition, we average pool the low-resolution branch of HRNet to produce a global feature vector and add a linear layer to predict an initialization Θ_0 for the parameters.

3.2. Feedback

Given an initial estimate, we reproject keypoints from the SMPL model to the feature maps F to retrieve spatial features. Each keypoint is only reprojected to one channel, yielding K channels for K keypoints. This design is motivated by model fitting, where the keypoint detector outputs one channel per keypoint.

For a reprojection $x_k = (u, v)$ where u and v are the pixel coordinates, we take the feature values inside a window centered at x_k as

$$f_k = \{f(x) \in F_k | \parallel x - x_k \parallel \leq r\} \quad (1)$$

where we set $r = 3$ pixels as the radius of the window. We concatenate feedback from all the keypoints, along with the current estimate Θ_t and the bounding box center c^{bbox} and scale s^{bbox}, to form the final feedback vector as

$$f = [f_1, ..., f_K, \Theta_t, c^{bbox}, s^{bbox}] \quad (2)$$

Types of Keypoints. The per-keypoint feature map does not directly detect a keypoint. Instead, each channel learns the features associated with a keypoint. This process does not require direct supervision, which allows us to test different types of keypoints where 2D annotations are unavailable. We test three types (Figure 3): semantic keypoints (K = 24); mocap markers (K = 67); and evenly sampled mesh vertices (K = 231). We examine the three types separately and do not combine them.

Of the three, mocap markers provide better pose and shape information than semantic keypoints, and are less redundant than mesh vertices. We use the same mocap markers as in AMASS [45], and the markers are defined on the mesh surface by selecting the closest vertices.

Full-frame Adjusted Reprojection. Human regression methods take a square-cropped image of a person as input, assuming the optical axis going through the crop center. But the input is usually cropped from a full-frame image with a different optical axis. This deviation incurs an error in the global rotation estimation, and by implication, the body pose estimation.

CLIFF [38] proposes to supervise the 2D reprojection loss in the original full-frame image, by using the full-frame reprojection:

$$x^{full}_{2D} = \Pi(X^{full}_{3D}) = \Pi(X_{3D} + t^{full}) \quad (3)$$

where Π is the perspective projection using the original camera intrinsics, X_{3D} are the body keypoints in the canonical body coordinate, and t^{full} is the translation with respect to the optical center of the original image. The camera intrinsics, if unknown, can be estimated from the dimensions of the full image [32]. This reprojection is more faithful to the image formation process and leads to a better global rotation estimation.

We propose to adjust the points back to the cropped image after full-frame reprojection:

$$x_{2D}^{crop} = (x_{2D}^{full} - c^{bbox})/s^{bbox} \tag{4}$$

where c^{bbox} and s^{bbox} are the location and the size of the bounding box from which the crop is obtained. We call this *full-frame adjusted reprojection*. This seemingly trivial operation grants two advantages. First, the scale of the reprojection is normalized. The reprojection error is unaffected by the size of the person in the original image.

But more importantly, it extends full-frame reprojection to the feedback step. We reproject keypoints to the cropped image feature maps, but with this camera model, the locations are properly adjusted to be consistent with full-frame reprojection. We use this model to retrieve f_k for each keypoint as in Eq 1.

Feedback Dropout. Each keypoint reprojection produces a feedback signal. To combine signals from all the keypoints, we formulate it as ensemble learning using dropout [56]. Specifically, we add a dropout layer, where there is a $p = 0.25$ chance that feedback f_k will be zeroed out during training.

During training, the network learns to infer the pose and shape using subsets of keypoints. This prevents co-adaptations [23] of feedback signals, and makes the network robust at test time when some keypoints are occluded.

This design also has ties to Spatial Dropout [60], which randomly drops feature map channels as opposed to pixel-wise activations. Similarly, we drop a keypoint's feedback completely instead of dropping some values.

3.3. Update

The update module takes the feedback signal f as input (Eq. 2), and predicts an update step $\Delta\Theta_t$, which is added to produce the next estimate: $\Theta_t + \Delta\Theta_t \rightarrow \Theta_{t+1}$.

The update is split into $N = 26$ parallel streams. Each stream is responsible for updating one SMPL parameter. There are 26 streams: 24 for the joint rotations, one for the shape parameter, and one for the translation.

Each stream has an identical structure, consisting of a GRU and a 2-layer MLP. The GRU updates its hidden state: $(f, h_{t-1}^n) \rightarrow h_t^n$, and the MLP maps the hidden state to the parametric update: $(h_t^n) \rightarrow \Delta\theta_t^n$. The parallel streams can be implemented as batch matrix operations to run efficiently.

The update module acts as a first-order optimizer. Each stream learns the update rule of a parameter. Multiple streams effectively disentangle the update rules. The feedback signal is similar to the data term that informs the error. The prior, which is hand-crafted in optimization, is now learned by the update module.

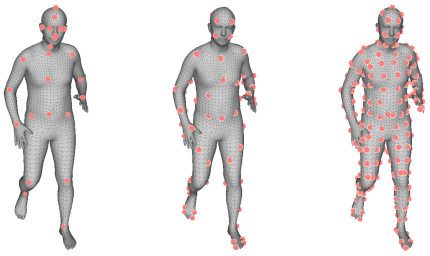

Figure 3. **Types of keypoints.** From left to right are semantic keypoints, mocap markers, and uniformly sampled vertices. We use one of the three types during feedback (Sec. 3.2).

Having 26 update streams does not increase the complexity because we also reduce the hidden units. The complexity per layer is $O(NM^2)$, where N is the number of streams and M is the size of the hidden layer. Previous works use $N = 1$ and $M = 1024$, leading to 1M parameters per layer. We use $N = 26$ and $M = 32$, which are 27K parameters per layer.

3.4. Supervision

The iterative update steps produce a sequence of estimates $\{\Theta_0, ..., \Theta_T\}$. At inference time, Θ_T is the final prediction. During training, we supervise all the iterations.

At each iteration t, the loss is made up of three terms,

$$\mathcal{L}_t = \lambda_{2D}\mathcal{L}_{2D}^t + \lambda_{3D}\mathcal{L}_{3D}^t + \lambda_{SMPL}\mathcal{L}_{SMPL}^t \tag{5}$$

where each term is calculated as

$$\begin{aligned} \mathcal{L}_{2D}^t &= ||\hat{\mathcal{J}}_{2D} - \Pi(\mathcal{J}_{3D}^t)||_F^2 \\ \mathcal{L}_{3D}^t &= ||\hat{\mathcal{J}}_{3D} - \mathcal{J}_{3D}^t||_F^2 \\ \mathcal{L}_{SMPL}^t &= ||\hat{\Theta} - \Theta_t||_2^2 \end{aligned}$$

$\mathcal{J}_{3D}$ are the 3D joints obtained from the SMPL model, the hat operator denotes the ground truth of that variable, and Π is the full-frame adjusted reprojection.

The final loss is a weighted sum of the loss at each iterative update

$$\mathcal{L} = \sum_{t=0}^{T} \gamma^{T-t}\mathcal{L}_t \tag{6}$$

where we set $\gamma = 0.85$ and $T = 5$ for all experiments. Prior works supervise only the last iteration to prevent the overshoot behavior, but it requires the gradient to flow through a long sequence of estimates, which slows down training convergence.

Our supervision is inspired by RAFT [58], an iterative optical flow method. First, the gradient is backpropagated through $\Delta\Theta_t$ but not through Θ_t. In other words, at each

iteration, we only supervise the update but not the prediction from the previous step. Second, we use the proposed weighted sum that downscales the importance of earlier iterations.

3.5. Applications

ReFit is trained with single-view images for single-view inference. But since ReFit operates similarly to an optimizer, we demonstrate two applications in which traditionally an optimizer is used: multi-view model fitting, and fitting a pre-acquired shape to images of the same subject.

Multi-view ReFit. Motion capture with multiple calibrated cameras offers the highest accuracy [18, 27]. In a markerless setting, an optimization procedure detects keypoints in each view and fits the model by minimizing the reprojection error [1]. We replace this procedure using ReFit as shown in Figure 4.

ReFit operates on each view independently to produce updates. The updates are averaged across views with a multi-view averaging procedure. We average the shapes by taking the mean of the predicted PCA coefficients across views. Because the body pose consists of the relative rotation of each joint, we directly average the body poses across views with rotation averaging, which averages the rotation matrices and uses SVD to project the result back to SO(3). The predicted global rotations are with respect to each view, so we transform them to a global coordinate with the calibrated extrinsic before performing rotation averaging. This procedure results in a single copy of the updated model. The model is then reprojected to all views to repeat the feedback-update step.

The difference between multi-view ReFit versus simply averaging the final predictions is that the averaging happens during fitting. So at each iteration, the update is in the direction that explains multiple views.

Shape ReFit. In this setting, we aim to fit a known body shape to images of the same subject with different poses. This is relevant for applications where the shape of a subject can be pre-scanned or pre-fitted and then used for outdoor motion capture (mocap) [21, 61]. Registering the shape to its images is commonly done with optimization, but we demonstrate that ReFit can perform a similar function.

We call this procedure Shape ReFit. At each iteration, we ignore the shape prediction and use the known shape to render the SMPL model. Therefore, the reprojection reflects the alignment status of the ground truth shape with the image, and ReFit will adjust the pose and position of the model for better alignment.

We perform experiments to verify that Shape ReFit can indeed fit a known and fixed shape to its images.

4. Experiments

Datasets. We train the ReFit model with 3DPW [61], Human3.6M [27], MPI-INF-3DHP [46], COCO [40] and MPII [4]. We use the pseudo ground truth SMPL annotations from EFT [29] for COCO and MPII. We evaluate ReFit on 3DPW and Human3.6M, with the MPJPE (mean per-joint error) and PA-MPJPE (Procrustes-aligned) metrics.

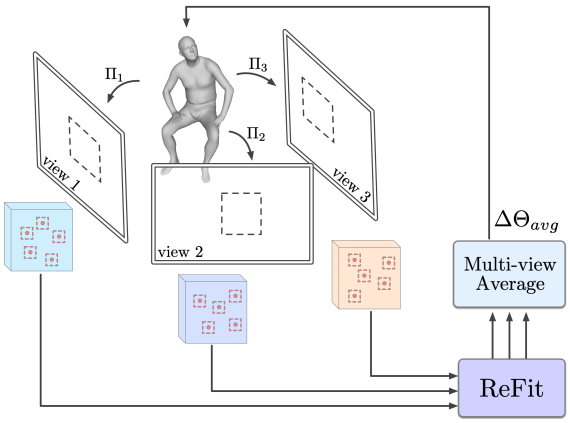

Figure 4. **Multi-view ReFit.** At each iteration, ReFit operates on all views independently to produce updates. The multi-view averaging procedure pool updates across views to produce a single update for the mesh. The updated mesh is again reprojected to each view to repeat the iterative process (Sec. 3.5).

Implementation. We train ReFit for 50k iterations with all the datasets and evaluate on 3DPW. We then finetune the model with only Human3.6M for another 50k iterations and evaluate on Human3.6M. We use the Adam optimizer with a learning rate of 1e-4 and a batch size of 64. The input image is resized to 256×256. Standard data augmentations are applied. We include additional details in the supplementary.

4.1. Quantitative Evaluation

We present quantitative evaluation in Table 1. Our method outperforms all single-view methods.

We organize the methods by their training data to present a moderate view of progress: the quality of training data plays a major role in improving test accuracy, as is pointed out by recent studies [50]. We follow this practice and train CLIFF [38] as our baseline with the same data and procedure. Overall, ReFit outperforms other architectures.

We exam the generalization by training with synthetic data from BEDLAM [7] following their proposed losses. Table 2 shows that ReFit generalizes better, and by using both synthetic and real data for training, it achieves the best results.

4.2. Ablations

We conduct ablation experiments to examine the core components of ReFit. All models are trained with the same data as the proposed main model, and tested on 3DPW.

Type of Keypoints. We train three models with the three types of keypoints in the feedback step. They are used for

Training	Architecture	3DPW		Human3.6M	
		MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
H+M+2D	HMR [30]	130.0	76.7	88.0	56.8
H+M+ 2D-SPIN [35]	HMR†	98.5	60.9	64.8	43.7
	ProHMR [36]	-	59.8	-	41.2
	PyMAF [67]	92.8	58.9	57.7	40.5
H+M+ 2D-EFT [29]	PARE* [34]	82.0	50.9	-	-
	HMR*†	79.5	48.0	58.8	39.5
	PyMAF* [66]	78.0	47.1	54.2	37.2
	ReFit*	**71.0**	**43.9**	**48.5**	**32.4**
H+M+ 3DPW+ 2D-EFT [29]	PARE* [34]	74.5	46.5	-	-
	PyMAF* [66]	74.2	45.3	-	-
	HybrIK [37]	74.1	45.0	55.4	33.6
	CLIFF*‡ [38]	73.5	44.3	52.6	35.0
	ReFit*	**65.8**	**41.0**	**48.4**	**32.2**

Table 1. **Evaluation** grouped by training data (H: H36M, M: MPI-INF-3DHP, 2D-X: 2D datasets with 3D pseudo-gt from method X). The superscripts denote ($*$: using HRNet backbone, †: implementation from Zhang et al. [66, 67], ‡: our implementation).

Training	Architecture	3DPW		
		MPJPE	PA-MPJPE	PVE
BEDLAM [7]	BEDLAM-CLIFF	72.0	46.6	85.0
	ReFit	**66.2**	**43.8**	**80.1**
BEDLAM + Real	**ReFit**	**57.6**	**38.2**	**67.6**

Table 2. **Evaluation**: with additional synthetic data from BEDLAM.

reprojection to query features, but there is no supervision on the 2D locations. Using mocap markers produces the best results, confirming previous studies [45, 68] that demonstrate mocap markers as a good proxy to infer the human body.

Feedback Dropout. We test the effect of dropout in the feedback step. We see that feedback dropout significantly boosts accuracy, likely because it prevents co-adaptations of keypoint signals, and makes the model robust when keypoints are occluded.

Full-frame Adjusted Reprojection. Our result confirms that using full-frame reprojection for supervision improves accuracy [38]. The proposed full-frame adjusted model extends to the feedback step. The best result is achieved when the camera model is faithful to the full image formation in all stages of the network.

Feedback Radius. For each keypoint, we query a window at its reprojection location as the feedback feature. The radius of the window affects the local context. The motivation is that the window encodes first-order information to indicate where the keypoint should move on the 2D image plane. Overall, we find that $r = 3$ works well.

Inference Iterations. We train the model with $T = 5$ update steps during training. At inference time, we test the model accuracy with various update steps. $T = 0$ indicates the initial guess from the backbone. Most previous methods

Experiment	Method	3DPW	
		MPJPE	PA-MPJPE
Type of Keypoints Reprojection	Semantic Keypoints	70.3	42.1
	Mocap Markers	65.8	41.0
	Sparse Vertices	69.2	41.9
Feedback Dropout	No Dropout	70.2	42.1
	$p = 0.15$	68.5	41.5
	$\boldsymbol{p = 0.25}$	65.8	41.0
Full-frame Adjusted Reprojection	No Full-frame	70.6	42.6
	Only Supervision	68.8	42.5
	Supervision+Feedback	65.8	41.0
Feedback Radius	$r = 0$	68.7	42.4
	$r = 1$	68.6	41.7
	$\boldsymbol{r = 3}$	65.8	41.0
Inference Iterations	$T = 0$	73.0	45.0
	$T = 2$	67.6	42.1
	$\boldsymbol{T = 5}$	65.8	41.0
	$T = 10$	66.9	42.0
Update Module	One GRU	69.6	42.3
	26 GRUs	65.8	41.0
Supervision	Last Iteration	70.5	44.3
	All Iterations	65.8	41.0

Table 3. **Ablation of model designs**. The highlighted option is used for the final model. We detail each experiment in Sec. 4.2.

use 3 steps, corresponding to $T = 2$ in our case. We see the benefit of using more update steps. We also observe that increasing to $T = 10$ does not cause the estimation to diverge.

Update Module. We use 26 parallel GRU streams to predict updates. To test the alternative, we swap out the 26 GRU with one larger GRU with 516 hidden units. This model has lower accuracy, confirming our hypothesis that it is beneficial to have separate update operators to learn separate update rules.

Supervision. We test supervising all iterations against supervising only the last iteration. Supervising all iterations achieves higher accuracy, but stopping the gradient across iterations as stated in Sec. 3.4 is important for stable training.

4.3. Qualitative Evaluation

We show qualitative results on 3DPW in Figure 5. We organize the results by MPJPE percentile, with X^{th} percentile indicating higher error than $X\%$ of the samples.

Overall, we observe accurate reconstructions with good alignment to images throughout different percentiles. We carefully inspect examples at the 99^{th} percentile and find that most examples have severe occlusions. Occlusion by another human produces a second level of complexity. We include more examples in the supplementary.

We provide examples from COCO in Figure 9 that highlights the difference between the initial estimation without refinement (T=0) and the final results from ReFit (T=5).

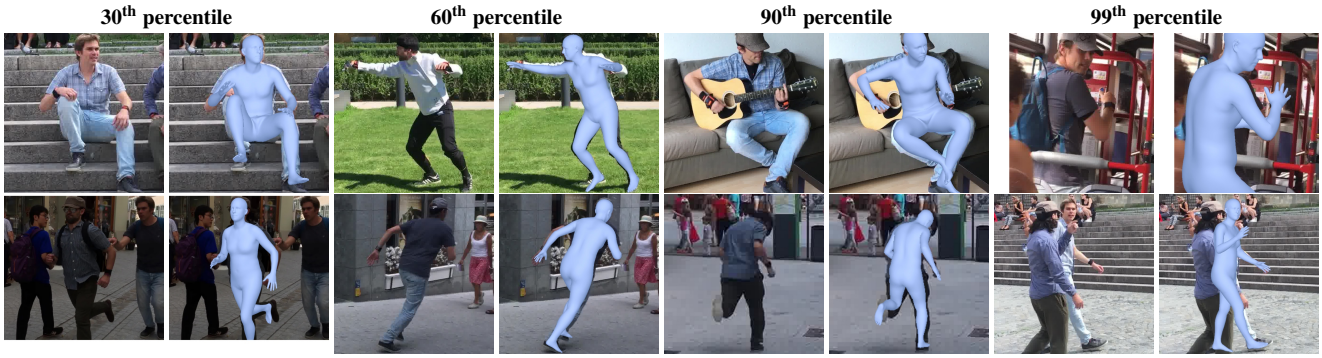

Figure 5. **Qualitative results from ReFit** on 3DPW. Examples are organized by MPJPE percentile. Higher percentile indicates higher error. For example, samples at the 99th percentile have higher error than 99% of the examples. The MPJPE at the four percentiles are 50.5mm, 65.4mm, 99.3mm and 158.8mm respectively. Samples at the 99th percentile often have severe occlusions or cropping.

Figure 6. **Shape ReFit.** From left to right are image, ReFit, ReFit with ground truth shape substitute, and Shape ReFit. White boxes highlight misalignment if shape is substituted in place.

Method	3DPW		Human3.6M	
	MPJPE	PA-MPJPE	MPJPE	PA-MPJPE
ReFit	65.8	41.0	48.4	32.2
ReFit + gt shape	**61.7**	**40.6**	**41.8**	**29.8**
Shape ReFit	**61.7**	40.7	44.4	32.1

Table 4. **Shape ReFit** on 3DPW and Human3.6M. Shape ReFit (Sec. 3.5) recovers more accurate poses than ReFit. Substituting shape in place (ReFit + gt shape) has similar accuracy, but produces misalignment (Fig. 6).

Method	Human3.6M		
	MPJPE	PA-MPJPE	PVE
ReFit	52.6	34.9	66.3
ReFit + avg	41.7	29.0	54.4
Multi-view ReFit (5 iters)	38.4	**26.6**	50.0
Multi-view ReFit (10 iters)	**37.5**	26.9	**48.4**

Table 5. **Multi-view ReFit** on Human3.6M. Per-vertex error (PVE) is computed with mesh recovered from MoSH [42]. Multi-view ReFit uses multi-view information during fitting, and is more accurate than simply averaging the predictions (ReFit + avg).

4.4. Application Evaluation

We evaluate how ReFit performs in the two proposed applications: Shape ReFit and Multi-view ReFit.

Shape ReFit is more accurate than ReFit, as ground truth shape is used during the fitting process (Tab 4). Replacing the final prediction from ReFit with the ground truth shape (ReFit + gt shape) yields similar pose accuracy. However, substituting the shape in place produces misalignment to the image when the predicted shape is noticeably different from the ground truth, as shown in Figure 6. From the qualitative examples, we see that Shape ReFit can indeed fit a pre-acquired shape to its image.

In the multi-view experiments, we run ReFit and Multi-view ReFit on S9 and S11 from Human3.6M (Tab 5). In the first baseline, we run ReFit on each view separately and report the average reconstruction errors. In the second baseline, we run ReFit on each view separately but average the final predictions. Averaging the predictions from multiple views improves the results, particularly in MPJPE, due to a better global rotation estimation. The proposed multi-view ReFit produces the best results. The improvement is more evident in terms of the per-vertex error (PVE), where we see a 27% improvement over the baseline. In this setting, we also see the benefit of running for more update iterations, with 10 iterations slightly better than 5 iterations.

We show qualitative results for Multi-view ReFit in Figure 7. We compare the results against the mesh recovered using MoSH [42]. Multi-view ReFit produces accurate 3D poses. We further test combining Multi-view with Shape ReFit. The recovered mesh is very close to MoSH results. MoSH is a proprietary optimization procedure that uses multi-view video and localized 3D markers, while our method only assumes multi-view images and optionally a pre-fitted shape. This result points to an alternative mocap procedure, where one can pre-fit the shape to an A-posed subject with optimization and use Multi-view + Shape ReFit for motion capture. The result can be used on its own or as initialization for optimization. A detailed comparison with MoSH is left for future work.

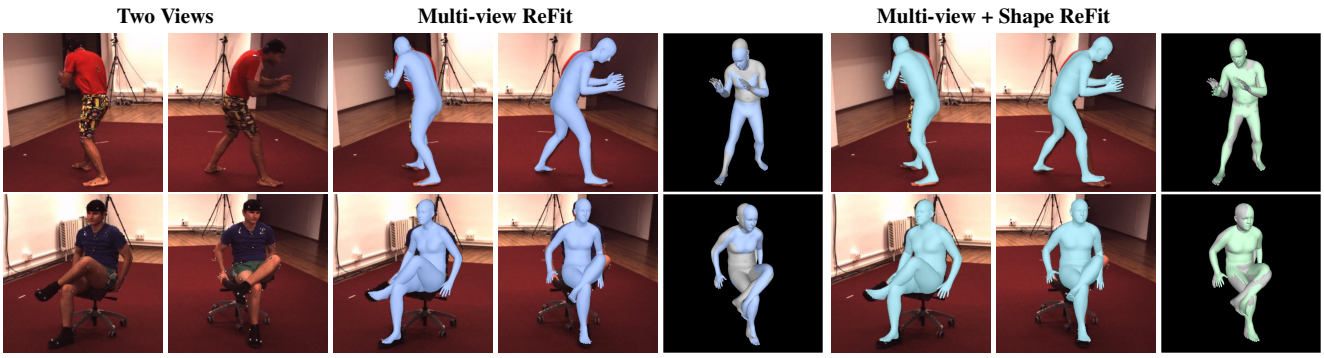

Figure 7. **Multi-vew ReFit.** We show results for 2 of 4 views. In the middle (blue meshes), multi-view ReFit reconstructs accurate poses compare to the ground truth grey meshes from MoSH [42]. On the right (green meshes), assuming ground truth shapes are pre-fitted and available, multi-view ReFit produces results that are very close to MoSH.

Figure 8. **Per-keypoint feature map.** Each channel corresponds to the learned features of a keypoint. Here we show feature maps from the semantic keypoint model. The network learns a peak (blue) or a valley (yellow surrounded by blue) response around the keypoint, without direct supervision.

Figure 9. **Refinement in-the-wild.** Examples from the coco validation set. From top to bottom are images, predictions without refinement update (T=0), and predictions from ReFit (T=5).

4.5. Visualization of Feature Maps

Feature maps from the proposed model are slightly blurry and harder to visualize, as they have positive and negative values. We train an alternative model, where we add a ReLU operator on the feature maps. This modification decreases the performance slightly, as the feature maps become less expressive (strictly positive), but makes them easier to visualize. Figure 8 shows examples from the model with ReLU, and we include more examples from both models in the supplementary.

We show feature maps from the semantic keypoint model, as semantic keypoints are more interpretable than markers. The model learns meaningful features around the corresponding keypoints, often as peaks or valleys. When a keypoint is reprojected onto the feature map, the peak or valley provides directional information that helps the next network layer infer where the keypoint should move on the 2D plane. This signal, combined with signals from other keypoints, can then be converted to updates for the 3D pose.

5. Conclusion

We have presented ReFit, the Recurrent Fitting Network that iteratively fits the parametric human model to images to recover the pose and shape of people. ReFit mirrors traditional model fitting, but learns the objective and update rules from data end-to-end. ReFit utilizes parallel GRU units that disentangle the learned update rules. Moreover, we demonstrate ReFit as a learned optimizer for multi-view fitting and shape fitting. ReFit is efficient to train and achieves state-of-the-art accuracy in challenging datasets.

Acknowledgements: We gratefully acknowledge support by the NSF grants FRR-2220868 and IIS-RI-2212433.

References

[1] Easymocap - make human motion capture easier. Github, 2021. 2, 5

[2] Jonas Adler and Ozan Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017. 2

[3] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018. 2

[4] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693, 2014. 5

[5] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416. 2005. 2

[6] Anurag Arnab, Carl Doersch, and Andrew Zisserman. Exploiting temporal context for 3d human pose estimation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3395–3404, 2019. 2

[7] Michael J Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. Bedlam: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8726–8737, 2023. 5, 6

[8] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 2

[9] 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 *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pages 561–578. Springer, 2016. 1, 2

[10] Christoph Bregler and Jitendra Malik. Tracking people with twists and exponential maps. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pages 8–15. IEEE, 1998. 2

[11] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299, 2017. 2

[12] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742, 2016. 2

[13] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014. 2

[14] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1964–1973, 2021. 2

[15] 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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16*, pages 769–787. Springer, 2020. 2

[16] Vasileios Choutas, Federica Bogo, Jingjing Shen, and Julien Valentin. Learning to fit morphable models. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VI*, pages 160–179. Springer, 2022. 2

[17] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2

[18] Junting Dong, Wen Jiang, Qixing Huang, Hujun Bao, and Xiaowei Zhou. Fast and robust multi-person 3d pose estimation from multiple views. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7792–7801, 2019. 2, 5

[19] Taosha Fan, Kalyan Vasudev Alwala, Donglai Xiang, Weipeng Xu, Todd Murphey, and Mustafa Mukadam. Revitalizing optimization for 3d human pose and shape estimation: A sparse constrained formulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11457–11466, 2021. 2

[20] John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. Deepview: View synthesis with learned gradient descent. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2367–2376, 2019. 2

[21] Marc Habermann, Weipeng Xu, Michael Zollhofer, Gerard Pons-Moll, and Christian Theobalt. Deepcap: Monocular human performance capture using weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5052–5063, 2020. 5

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[23] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012. 4

[24] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[25] David Hogg. Model-based vision: a program to see a walking person. *Image and Vision computing*, 1(1):5–20, 1983. 2

[26] Yinghao Huang, Federica Bogo, Christoph Lassner, Angjoo Kanazawa, Peter V Gehler, Javier Romero, Ijaz Akhter, and Michael J Black. Towards accurate marker-less human shape

and pose estimation over time. In *2017 international conference on 3D vision (3DV)*, pages 421–430. IEEE, 2017. 2

[27] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. 2, 5

[28] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5579–5588, 2020. 2

[29] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3d human model fitting towards in-the-wild 3d human pose estimation. In *2021 International Conference on 3D Vision (3DV)*, pages 42–52. IEEE, 2021. 5, 6

[30] Angjoo Kanazawa, Michael J Black, David W Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7122–7131, 2018. 1, 2, 6

[31] Angjoo Kanazawa, Jason Y Zhang, Panna Felsen, and Jitendra Malik. Learning 3d human dynamics from video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5614–5623, 2019. 2

[32] Imry Kissos, Lior Fritz, Matan Goldman, Omer Meir, Eduard Oks, and Mark Kliger. Beyond weak perspective for monocular 3d human pose estimation. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 541–554. Springer, 2020. 4

[33] Muhammed Kocabas, Nikos Athanasiou, and Michael J Black. Vibe: Video inference for human body pose and shape estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5253–5263, 2020. 2

[34] Muhammed Kocabas, Chun-Hao P Huang, Otmar Hilliges, and Michael J Black. Pare: Part attention regressor for 3d human body estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11127–11137, 2021. 2, 6

[35] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2252–2261, 2019. 1, 2, 6

[36] Nikos Kolotouros, Georgios Pavlakos, Dinesh Jayaraman, and Kostas Daniilidis. Probabilistic modeling for human mesh recovery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11605–11614, 2021. 6

[37] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3383–3393, 2021. 6

[38] 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 *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 590–606. Springer, 2022. 3, 5, 6

[39] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1954–1963, 2021. 2

[40] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[41] Lahav Lipson, Zachary Teed, Ankit Goyal, and Jia Deng. Coupled iterative refinement for 6d multi-object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6728–6737, 2022. 2

[42] Matthew Loper, Naureen Mahmood, and Michael J Black. Mosh: motion and shape capture from sparse markers. *ACM Trans. Graph.*, 33(6):220–1, 2014. 7, 8

[43] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015. 1, 2, 3

[44] Zhaoyang Lv, Frank Dellaert, James M Rehg, and Andreas Geiger. Taking a deeper look at the inverse compositional algorithm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4581–4590, 2019. 2

[45] Naureen Mahmood, Nima Ghorbani, Nikolaus F Troje, Gerard Pons-Moll, and Michael J Black. Amass: Archive of motion capture as surface shapes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5442–5451, 2019. 3, 6

[46] 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 *2017 international conference on 3D vision (3DV)*, pages 506–516. IEEE, 2017. 5

[47] Lea Muller, Ahmed AA Osman, Siyu Tang, Chun-Hao P Huang, and Michael J Black. On self-contact and human pose. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9990–9999, 2021. 2

[48] Markus Oberweger, Paul Wohlhart, and Vincent Lepetit. Training a feedback loop for hand pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3316–3324, 2015. 2

[49] Mohamed Omran, Christoph Lassner, Gerard Pons-Moll, Peter Gehler, and Bernt Schiele. Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In *2018 international conference on 3D vision (3DV)*, pages 484–494. IEEE, 2018. 2

[50] Hui En Pang, Zhongang Cai, Lei Yang, Tianwei Zhang, and Ziwei Liu. Benchmarking and analyzing 3d human pose and shape estimation beyond algorithms. *Advances in Neural Information Processing Systems*, 35:26034–26051, 2022. 5

[51] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10975–10985, 2019. 2

[52] Georgios Pavlakos, Jitendra Malik, and Angjoo Kanazawa. Human mesh recovery from multiple shots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1485–1495, 2022. 2

[53] Davis Rempe, Tolga Birdal, Aaron Hertzmann, Jimei Yang, Srinath Sridhar, and Leonidas J Guibas. Humor: 3d human motion model for robust pose estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11488–11499, 2021. 2

[54] Cristian Sminchisescu and Alexandru C Telea. Human pose estimation from silhouettes. a consistent approach using distance level sets. In *10th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG'02)*, volume 10, 2002. 2

[55] Jie Song, Xu Chen, and Otmar Hilliges. Human body model fitting by learned gradient descent. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 744–760. Springer, 2020. 2

[56] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 4

[57] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5693–5703, 2019. 2, 3

[58] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 2, 4

[59] Garvita Tiwari, Dimitrije Antić, Jan Eric Lenssen, Nikolaos Sarafianos, Tony Tung, and Gerard Pons-Moll. Pose-ndf: Modeling human pose manifolds with neural distance fields. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*, pages 572–589. Springer, 2022. 2

[60] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 648–656, 2015. 4

[61] 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 *Proceedings of the European conference on computer vision (ECCV)*, pages 601–617, 2018. 1, 2, 5

[62] Alexander Weiss, David Hirshberg, and Michael J Black. Home 3d body scans from noisy image and range data. In *2011 International Conference on Computer Vision*, pages 1951–1958. IEEE, 2011. 2

[63] Donglai Xiang, Hanbyul Joo, and Yaser Sheikh. Monocular total capture: Posing face, body, and hands in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10965–10974, 2019. 2

[64] Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Ghum & ghuml: Generative 3d human shape and articulated pose models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6184–6193, 2020. 2

[65] Andrei Zanfir, Eduard Gabriel Bazavan, Mihai Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. Neural descent for visual 3d human pose and shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14484–14493, 2021. 2

[66] Hongwen Zhang, Yating Tian, Yuxiang Zhang, Mengcheng Li, Liang An, Zhenan Sun, and Yebin Liu. Pymaf-x: Towards well-aligned full-body model regression from monocular images. *arXiv preprint arXiv:2207.06400*, 2022. 2, 6

[67] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11446–11456, 2021. 1, 2, 6

[68] Yan Zhang, Michael J Black, and Siyu Tang. We are more than our joints: Predicting how 3d bodies move. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3372–3382, 2021. 6