

# On Self-Contact and Human Pose

## \*Supplementary Material\*

Lea Müller<sup>1</sup>, Ahmed A. A. Osman<sup>1</sup>, Siyu Tang<sup>2</sup>, Chun-Hao P. Huang<sup>1</sup>, Michael Black<sup>1</sup>

<sup>1</sup>MPI for Intelligent Systems, Tübingen, <sup>2</sup>ETH Zürich

{lea.mueller, ahmed.osman, stang, paul.huang, black}@tuebingen.mpg.de

The Supplementary Material provides additional details about our methods and visualizations of results.

### 1. Self-Contact Datasets

#### 1.1. 3D Contact Pose (3DCP) Meshes

##### 1.1.1 3DCP Scan

Raw scans have varying topology. To bring a corpus of scans to a common topology is the process of “registration”. Most traditional registration methods ignore interpenetration and self-contact. Registering our self-contact scans without modeling self-contact would result in self-penetration, particularly where the extremities contact the body. We address this by modifying the registrations objective function to encourage self-contact without penetration.

Specifically, the fitting objective includes a data term  $E_S$  evaluating the goodness of fit of the the vertices  $x_v$  on the template  $V$  to  $n$  randomly sampled points,  $x_s$  on the surface of the scan  $S$

$$E_S(S; V) = \frac{1}{n} \int_{x_s \in S} \rho(\|x_s - x_v\|) \quad (1)$$

where  $\rho$  is the Geman-McClure robust penalty function.

Additionally, we introduce a self-contact preserving energy term  $E_C$  to the objective function. The term  $E_C$  helps to minimize and preserve the *point-to-plane* distance between body parts that are in contact.  $E_C$  considers the set of contacting vertex pairs  $M_C$  defined by Definition 3.1 in the main paper. For each tuple  $(v_i, v_j)$  in  $M_C$ , we minimize the *point-to-plane* distance between triangles including  $v_i$  and the triangular planes including  $v_j$ . The contact energy term ensures that body parts that are in contact remain in contact.

The objective function is minimized in two steps: first a model fitting step, where it is minimized with respect to the SMPL-X model pose parameters  $\vec{\theta} \in \mathbb{R}^{55 \times 3}$  and body shape parameters  $\vec{\beta} \in \mathbb{R}^{25}$ . Following model fitting, a model-free optimization step minimizes point-to-plane distance between the model vertices  $x_v$  and the scan. A sample of the registrations is shown in Figure 1.

##### 1.1.2 3DCP Mocap.

**Sampling meshes from AMASS.** First, each mocap. sequence is sampled at half of its original frame rate. For each sampled mesh, we compute the contact signatures  $M_S$  with  $t_{eucl} = 3\text{cm}$ ,  $t_{geo} = 30\text{cm}$  and  $K = 98$ . The regions are visualized in Fig. 2. We select only one pose for each unique signature, while ignoring contact when it occurs in more than 1% of the data. We obtain a subset of 20,114 poses with unique self-contact signatures, as shown in Fig. 3.

**Self-Contact Optimization.** Here we provide details of the self-contact optimization for body meshes from the AMASS dataset. In this optimization, vertex pairs in  $M_C$  are further pulled together via a contact term  $\mathcal{L}_C$  and vertices inside the mesh are pushed to the surface via a pushing term  $\mathcal{L}_P$ , while  $\mathcal{L}_O$  ensures that vertices far away from contact regions stay in place. Note that  $\mathcal{L}_P$  and  $\mathcal{L}_C$  are slightly different from the loss terms in the main paper.  $\mathcal{L}_H$  is a prior for contact between hand and body and  $\mathcal{L}_A$  aligns the vertex normals when contact happens.

Given the set of vertices  $M_V$  of mesh  $M$ ,  $M_E \subset M_V$  denotes the subset of vertices affiliated with extremities,  $M_I \subset M_V$  denotes the subset of vertices inside the mesh, and  $M_{EI} = M_E \cap M_I$  denotes the vertices of extremities that are inside the mesh itself and  $M_{EI}^c$  its complement. We identify vertices inside the mesh using generalized winding numbers [1].  $M_{V_H} \subset M_V$  is the subset of hand vertices. Note that we make SMPL-X watertight by closing the back of the mouth.  $M_C$  is computed following Definition 3.1 in the main paper with  $t_{geo} = 30\text{cm}$  and  $t_{eucl} = 3\text{cm}$  and  $M_G(v_i) = \{v_j | geo(v_i, v_j) > t_{geo}\}$ . Given an initial mesh  $\tilde{I}$ , we aim to minimize the objective function

$$\mathcal{L}(\theta_b, \theta_{h_l}, \theta_{h_r}) = \lambda_C \mathcal{L}_C + \lambda_P \mathcal{L}_P + \lambda_H \mathcal{L}_H + \lambda_O \mathcal{L}_O + \lambda_A \mathcal{L}_A + \lambda_{\theta_h} \mathcal{L}_{\theta_h} \quad (2)$$

$$\lambda_{\theta} \mathcal{L}_{\theta}, \quad (3)$$



Figure 1. A representative sample from the registrations. A total of 3 male and 3 female subjects were scanned in a diversity of poses that involve self-contact. The 3D scans are registered to a common mesh topology by fitting the SMPL-X template mesh to them using a self-contact preserving energy term that penalizes body part interpenetration.

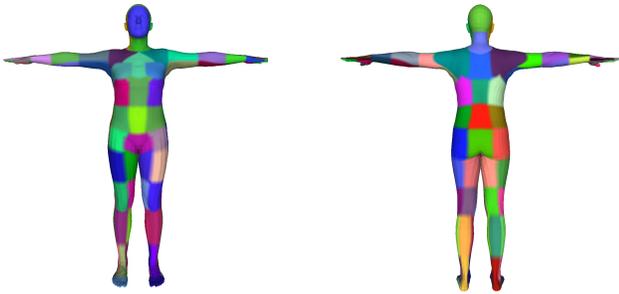


Figure 2. To compute self-contact signatures, we group vertices into distinct regions, shown here with different colors. This is useful for searching our scan datasets for poses with specific types of contact.



Figure 3. Sample poses from 20 unique contact signatures,  $\mathbf{S}$ . We apply  $\mathbf{S}$  to select interesting self-contact poses in AMASS.

where  $\theta_h$  denote the hand pose vector of the SMPL-X model. Further,

$$\mathcal{L}_C = \frac{1}{|M_{EI}^c|} \sum_{v_i \in M_{EI}^c} a \alpha \tanh\left(\frac{f_g(v_i)}{\alpha}\right),$$

$$\mathcal{L}_P = \frac{1}{|M_{EI}|} \sum_{v_i \in M_{EI}} \gamma_1 \tanh\left(\frac{f_g(v_i)}{\gamma_2}\right), \text{ and}$$

$$\mathcal{L}_H = \frac{1}{|M_{VH}|} \sum_{v_i \in M_{VH}} \delta_1 h_{v_i} \tanh\left(\frac{f_g(v_i)}{\delta_2}\right),$$

where  $f_g$  denotes a function, that for each vertex  $v_i$  finds the closest vertex in self contact  $v_j$ , or mathematically  $f_g(v_i) = \min_{v_j \in M_G(v_i)} \|v_i - v_j\|_2$ .  $h_{v_i}$  denotes the weight per hand vertex from the hand-on-body prior  $\mathcal{L}_H$  as explained below, if  $v_i$  is outside, otherwise  $h_{v_i} = 1$ . Further,  $a = (\min_{v_j \in \mathcal{U}(M_C)} \text{geo}(v_i, v_j) + 1)^{-1}$  is an attraction weight. This weight is higher, for vertices close to vertices in contact of  $\tilde{I}$ .  $\mathcal{L}_\theta$  is a  $L_2$  prior that penalizes deviation from the initial pose and  $\mathcal{L}_{\theta_h}$  defines an  $L_2$  prior on the left and right hand pose using the a low-dimensional hand pose space.  $\alpha = 0.04$ ,  $\gamma_1 = 0.07$ ,  $\gamma_2 = 0.06$  define slope and

offset of the pulling and pushing terms. For the hand-on-body-prior we use  $\delta_1 = 0.023$ , and  $\delta_2 = 0.02$  if  $v_i$  is inside and  $\delta_1 = \delta_2 = 0.01$  if  $v_i$  is outside the mesh.

Self-contact optimization aims to correct interpenetration and encourage near-contact vertices to be in contact by slightly refining the poses around the contact regions. Vertices that are not affected should stay as close to the original positions as possible. In  $\mathcal{L}_O$ , the displacement of each vertex from its initial position is weighted by its geodesic distance to a vertex in contact. Given  $\tilde{v}_i$  denoting the position of vertex  $i$  of  $\tilde{I}$ , the outside loss term is

$$\mathcal{L}_O = \delta_2 \sum_{v_i \in M_V} \min_{v_j \in \mathcal{U}(M_C)} \text{geo}(v_i, v_j)^2 \|v_i - \tilde{v}_i\|_2,$$

where  $\min_{v_j \in \mathcal{U}(M_C)} \text{geo}(v_i, v_j) = 1$  if  $M_C = \emptyset$  and  $\delta_2 = 4$ . Lastly, we use a term,  $\mathcal{L}_A$ , that encourages the vertex normals  $N(v)$  of vertices in contact to be aligned but in opposite directions:

$$\mathcal{L}_A = \frac{1}{|M_C|} \sum_{(v_i, v_j) \in M_C} 1 + \langle N(v_i), N(v_j) \rangle.$$

**Hand-on-Body Prior.** Hands and fingers play an important role as they frequently make contact with the body. However, they have many degrees of freedom, which makes their optimization challenging. Therefore, we learn a hand-on-body prior from 1279 self contact registrations. For this, we use only poses where the minimum point-to-mesh distance between hand and body is  $< 1\text{mm}$ . These are 718 and 701 poses for the right and left hand, respectively. Since left and right hand are symmetric in SMPL-X, we unite left and right hand poses. Across the 1429 poses, the mean distances per hand vertex to the body surface,  $d_m(v_i)$  ranges per vertex from 1.79 to 5.52 cm, as visualized in Fig. 4. To obtain the weights  $h_{v_i}$  in  $\mathcal{L}_H$ , we normalize  $d_m(v_i)$  to  $[0, 1]$ , denoted as  $s(d_m(v_i))$ , and obtain the vertex weight by  $h_{v_i} = -s(d_m(v_i)) + 1$ .

## 1.2. Mimic-The-Pose (MTP) Data

**AMT task details.** It can be challenging to mimic a pose precisely. To simplify the process for workers on AMT, we give detailed instructions, add thumbnails to compare the own image with the presented one and, most importantly, highlight the contact areas; see Fig. 5. To gain more variety, we also request that participants make small changes in the environment for each image, e.g. by rotating the camera, changing clothes, or turning lights on/off. We also ask participants to mimic the global orientation of the center image. For more variety in global orientation, we vary body roll from  $-90^\circ$  to  $90^\circ$  in  $30^\circ$  steps, resulting in seven different presented global orientations. For example, in the first and third row of Fig. 5, the center image shows the presented pose from a frontal view. In the second and fourth

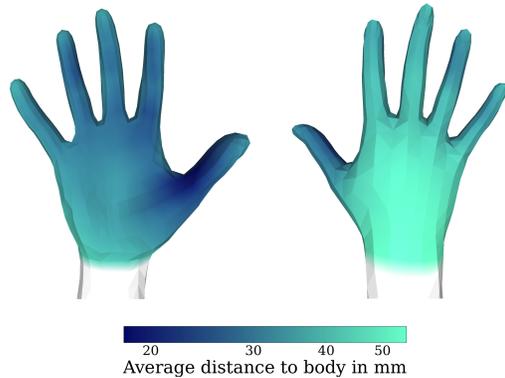


Figure 4. Hand on body prior. Dark blue indicates small distances to body on average across all registrations where hands are close to the body. The prior is identical for left and right hand.

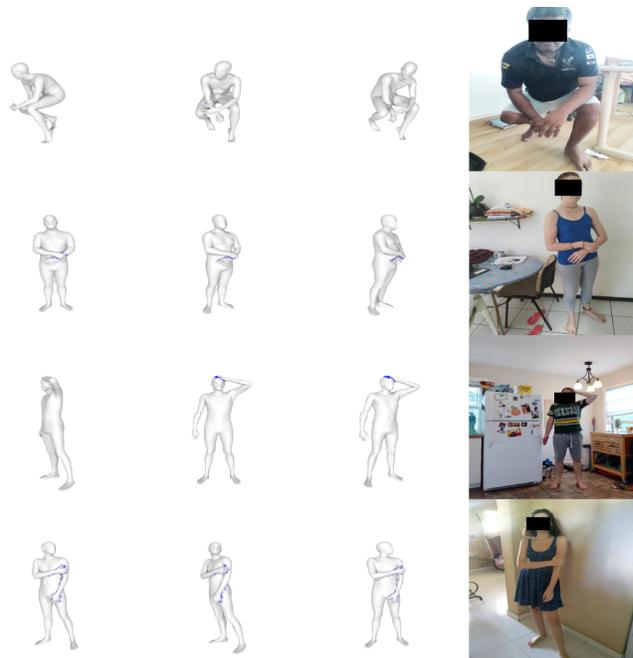


Figure 5. Presentation format and examples of mimicked poses from the MTP data set. On the left side, the presented pose with contact highlighted in blue. Humans mimicking the poses on the right.

row, the center body has different orientations. We also ask participants for their height, weight, and gender (M, F, and Non-Binary).

**SMPLify-XMC.** In the first stage, we optimize body shape  $\beta$  and camera  $\Pi$  (focal length, rotation and translation), and body global orientation  $\theta_g$ , using ground-truth height in meters,  $h_{gt}$ , and weight in kg,  $w_{gt}$ . The objective function of the first stage is given as

$$\mathcal{L}(\beta, \Pi, \theta_g) = \lambda_{\theta_g} \mathcal{L}_{\theta_g} + \lambda_M \mathcal{L}_M + E_J.$$

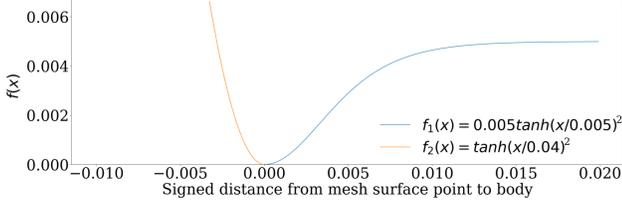


Figure 6. Functions to regulate the self-contact pushing and pulling term in SMPLify-XMC.  $f_1$  is used in  $\mathcal{L}_C$ ,  $f_2$  is used in  $\mathcal{L}_P$ . The parameters ensure that inside vertices are pushed out quickly, while vertices in contact are pulled together as long as they are close enough.

$\mathcal{L}_M = e^{100|M_h - h_{gt}|} + e^{|M_w - w_{gt}|}$  is the measurements loss, where  $M_h$  and  $M_w$  are height and weight of mesh  $M$ . We compute height and weight from mesh v-template in a zero pose (T-pose). For height, we compute the distance between the top of the head and the mean point between left and right heel. For weight, we compute the mesh volume and multiply it by  $985 \text{ kg/m}^3$ , which approximates human body density.  $\mathcal{L}_{\theta_g}$  is a loss that allows rotation around the y-axis, but not around x and z.

In Fig. 6 we visualize the pushing and pulling terms used in the SMPLify-XMC objective. We use 6 PCA components for the hand pose space and initialize the fitting with a mean hand pose. In contrast to SMPLify-X we do not ignore hip joints and double the joint weights for knees and elbows. Before optimization, we resize images and keypoints to a maximum height or width of 500 pixel. Similar to SMPLify-X we use the PyTorch implementation of fast L-BFGS with strong Wolf line search as the optimizer [5]. We do not use the VPoser pose prior for SMPLify-XMC because we have a strong prior from the presented pose.

We notice that the presented global orientation is not always mimicked well. For example, in row 4 of Fig. 5 the presented global orientation has a 60 degree rotation, whereas the mimicked image is taken from a frontal view. To better initialize the optimization, we select the best body orientation,  $\theta_g$ , among the seven presented ones based on their re-projection errors; then we compute the camera translation by again minimizing the re-projection error. We set the initial focal length,  $f_x$  and  $f_y$ , to 2170, which is the average of available EXIF data. These values, along with mean shape and presented pose are used to initialize the optimization.

In addition, SMPL and SMPL-X have not been trained to avoid self intersection. Therefore, we identify seven body segments that tend to intersect themselves, e.g. torso and upper arms (see Fig. 7). We test each segment for self intersection and thereby filter irrelevant intersections from  $M_I$ .

**MTP Dataset Details.** We sample meshes from 3DCP Scan, 3DCP Mocap., and AGORA [6] to comprise



Figure 7. Body segmented into regions where intersection can happen, since SMPL and SMPL-X are not trained to avoid self intersection. Per segment, we create closed meshes that allow for individual intersection tests. For self-contact, intersections that happen within a segment are not relevant. The hands are not included in any segment, because self intersections within hands or between hands and lower arm are not plausible and need to be resolved.

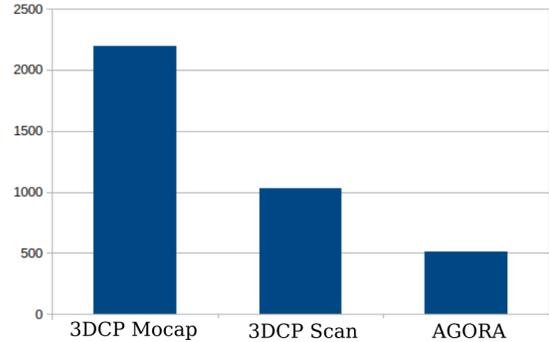


Figure 8. Image count in MTP Dataset per 3DCP subset.



Figure 9. Discrete self-contact can be challenging to annotate. Here we show a few example images that are annotated as having discrete self-contact between the left upper and lower arm (yellow circle). In the last two images, however, the upper and lower arm are barely touching. We do not consider these to be in self-contact. Another ambiguous case, this time due to occlusion, are the two legs in the first image. An annotator can only assume that the shin and calf are touching, based on semantic knowledge about human pose.

the presented meshes in MTP dataset. In total, we present 1653 different meshes, from which 1498 (90%) are contact poses following Definition 3.1 in the main document. Of the 1653 meshes, 110 meshes are from 3DCP Scan, 1304 meshes are from 3DCP Mocap., and 159 are from AGORA. We collect at least one image for each mesh. From the 3731 collected images, 3421 (92%) images show a person mimicking a contact pose. Figure 8 shows how many image we collected per subset.

### 1.3. Discrete Self-Contact (DSC) Data.

**Image selection.** Discrete self-contact annotation may be ambiguous and we find some annotations that we do not consider to be functional self-contact. For example, in

Fig. 9, some annotators label the left lower arm and left upper arm to be in contact, because of the slight skin touching at the elbow; we do not treat these as in self-contact. Therefore, we leverage the kinematic tree structure provided by SMPL-X and, in order to train TUCH, ignore the following annotations: left hand - left lower arm, left lower arm - left elbow, left lower arm - left upper arm, left elbow - left upper arm, left upper arm - torso, left foot - left lower leg, left lower leg - left knee, left lower leg - left upper leg, left knee - left upper leg, right hand - right lower arm, right lower arm - right elbow, right lower arm - right upper arm, right elbow - right upper arm, right upper arm - torso, right foot - right lower leg, right lower leg - right knee, right lower leg - right upper leg, right knee - right upper leg.

## 2. TUCH

Here we provide details of the SMPLify-XMC and SMPLify-DC methods and how we apply them on MTP and DSC data respectively.

SMPLify-XMC is explained in Sec. 4.2 of the main paper. It is applied, before the training, to all MTP images to obtain gender-specific pseudo ground-truth SMPL-X fits. To use these fits for TUCH training, two pre-processing steps are necessary. First, they are converted to neutral SMPL fits. Second, we transform the converted SMPL fits to the camera coordinate frame estimated during SMPLify-XMC. This is necessary since SPIN assumes an identity camera rotation matrix. After that, the data is treated as ground truth during training, which means we apply the regressor loss directly on the converted SMPL pose and shape parameters without in-the-loop fitting.

On the contrary, SMPLify-DC is applied during TUCH training to images with discrete self-contact annotations. We run 10 iterations of SMPLify-DC for each image in a mini batch.

MTP and the DeepFashion subset of DSC do not have ground-truth 2D keypoints but we find OpenPose detections good enough in both cases. For the 2D re-projection loss, we use ground-truth keypoints (if available) and OpenPose detections weighted by the detection confidence. Each mini batch consists of 50% DSC and 50% MTP data.

**Implementation details:** We initialize our regression network with SPIN weights [4]. We use the Adam optimizer [3] and a learning rate of  $1e - 5$ .

## 3. TUCH<sub>EX</sub>

One disadvantage of training with fitting in the loop is that it is relatively slow. As an alternative, we also explore Exemplar Fine-Tuning (EFT) [2], which is a regression based method for fitting 3D meshes to a single image. The fitted SMPL meshes may then be used as pseudo annotations to train a regressor without in-the-loop optimization.



Figure 10. RGB images from 3DCP Scan test set. A subject performing a pose with self-contact in a 3D body scanner.

With this approach, the authors train HMR-EFT, with which they achieve good results on 3DPW and MPI-INF-3DHP.

The idea of using discrete contact annotations is not limited to optimization based approaches. We show that they can also be applied in combination with EFT. Specifically, we extend the regressor loss of EFT with the contact terms from SMPLify-DC. We denote such an “EFT + contact loss” approach as EFT-C. Note that the original EFT loss uses a 2D orientation term to match the lower legs orientation, which we do not use here.

Each image in DSC is then paired with a pre-computed pseudo ground truth from EFT-C, and we denote the dataset as  $[DSC]_{EFT-C}$ . Then, we finetune the HMR-EFT network on MTP,  $[DSC]_{EFT-C}$ , as well as other training data from [2]. This new model is called TUCH with EXemplar Finetuning, TUCH<sub>EX</sub>. Unlike TUCH that still performs SMPLify-DC in the training loop, TUCH<sub>EX</sub> is supervised only by pre-computed fits so it can be trained faster.

**Implementation details.** We initialize our network with state-of-the-art HMR-EFT weights. We train TUCH<sub>EX</sub> on  $[COCO-All]_{EFT}$  (CAE), H36M, MPI-INF-3DHP (MI),  $[DSC]_{EFT-C}$ , and MTP.  $[COCO-All]_{EFT}$  denotes the COCO dataset after EFT processing, as described in [2]. In each batch we use a 10% CAE, 20% H36M, 10% MI, 20% 3DPW, 20%  $[DSC]_{EFT-C}$ , and 20% MTP. The remaining details are the same as in the TUCH implementation. For the DSC dataset, we only consider images where the full body is visible. To identify these images, we test whether the OpenPose detection confidence of ankles, hips, shoulders, and knees is  $\geq 0.2$ . We also ignore discrete contact annotations for connected body parts, as defined in 1.3.

## 4. Evaluation

**3DCP Scan test images.** During the scanning process when creating 3DCP Scan, we also take RGB photos of subjects being scanned, as shown in Figure 10. These images

Finetuning Data		MPJPE ↓				PA-MPJPE ↓			
		contact	no contact	unclear	total	contact	no contact	unclear	total
HMR-EFT [2]	-	88.3	84.6	83.6	85.3	52.1	<b>53.3</b>	<b>48.5</b>	<b>51.7</b>
TUCH <sub>EX</sub>	CAE + H36M + MI + 3DPW + [DSC] <sub>EFT-C</sub> + MTP	<b>82.8</b>	<b>83.2</b>	<b>80.3</b>	<b>82.3</b>	<b>50.4</b>	54.1	48.7	<b>51.7</b>

Table 1. Evaluation of TUCH<sub>EX</sub> for contact classes. CAE = [COCO-All]<sub>EFT</sub> as denoted in [2]. Bold numbers indicate the better a result.

	MPJPE ↓	PA-MPJPE ↓
SPIN	96.9	59.2
TUCH (MTP)	88.7	57.4
TUCH (MTP+DSC)	<b>84.9</b>	<b>55.5</b>

Table 2. Ablation of MTP data and DSC data.

	MPJPE ↓		PA-MPJPE ↓	
	3DPW	MI	3DPW	MI
HMR-EFT [2]	85.3	105.3	<b>51.7</b>	68.4
TUCH <sub>EX</sub>	<b>82.3</b>	<b>101.5</b>	<b>51.7</b>	<b>66.4</b>

Table 3. Training with 3DPW. Evaluation on 3DPW and MPI-INF-3DHP (MI). We report MPJPE and PA-MPJPE for different subsets of our data set.

have high-fidelity ground-truth poses and shapes from the registration process described in Sec. 1.1.1, making them a good test set for evaluation purposes. It is worth noting again that TUCH has never seen these images or subjects, but the contact poses were mimicked in creation of MTP, which is used in training TUCH.

**TUCH.** In Fig. 11 we visualize the improvement of TUCH over SPIN qualitatively. One can see that TUCH reconstructs bodies with better self-contact and less interpenetration (row 1 and row 2). Fig. 12, on the other hand, shows examples where SPIN is better than TUCH. Four of the images in Fig. 12 do not show the full body (rows 3, 4, 5, and 8). A possible reason why SPIN is better than TUCH in these cases is that MTP images always show the full body of a person, thus TUCH could be more sensitive to occlusion than SPIN.

We also evaluate the contribution of MTP data by fine-tuning SPIN only with it. The results are reported in Table 2, where TUCH (MTP+DSC) is the same as reported in Table 3 of the main paper. This experiment shows that MTP data alone is already sufficient to significantly improve state-of-the-art (SOTA) methods on 3DPW benchmarks. This suggests that the MTP approach is a useful new tool for gathering data to train neural networks.

**TUCH<sub>EX</sub>.** For an additional comparison with SOTA EFT [2], we evaluate our TUCH<sub>EX</sub> model on the same datasets (3DPW, MPI-INF-3DHP (MI), and 3DCP Scan)

	MPJPE ↓	PA-MPJPE ↓	MV2VE ↓
HMR-EFT [2]	<b>71.4</b>	48.3	83.9
TUCH <sub>EX</sub>	73.4	<b>43.3</b>	<b>82.8</b>

Table 4. Evaluation on 3DCP Scan test images. We report MPJPE, PA-MPJPE, and MV2VE.

with error measures (MPJPE, PA-MPJPE, and MV2VE) like TUCH, see Tables 1, 3, and 4.

The MPJPE of TUCH<sub>EX</sub> improves over HMR-EFT when evaluated on 3DPW. PA-MPJPE improves for contact poses and is overall on-par. Also the results on MPI-INF-3DHP improve. For the 3DCP Scan test set, PA-MPJPE improves. This shows that our data can not only be used with optimization based approaches, but also with exemplar fine-tuning, and that it allows us to improve the latest models in terms of estimating poses with contact.

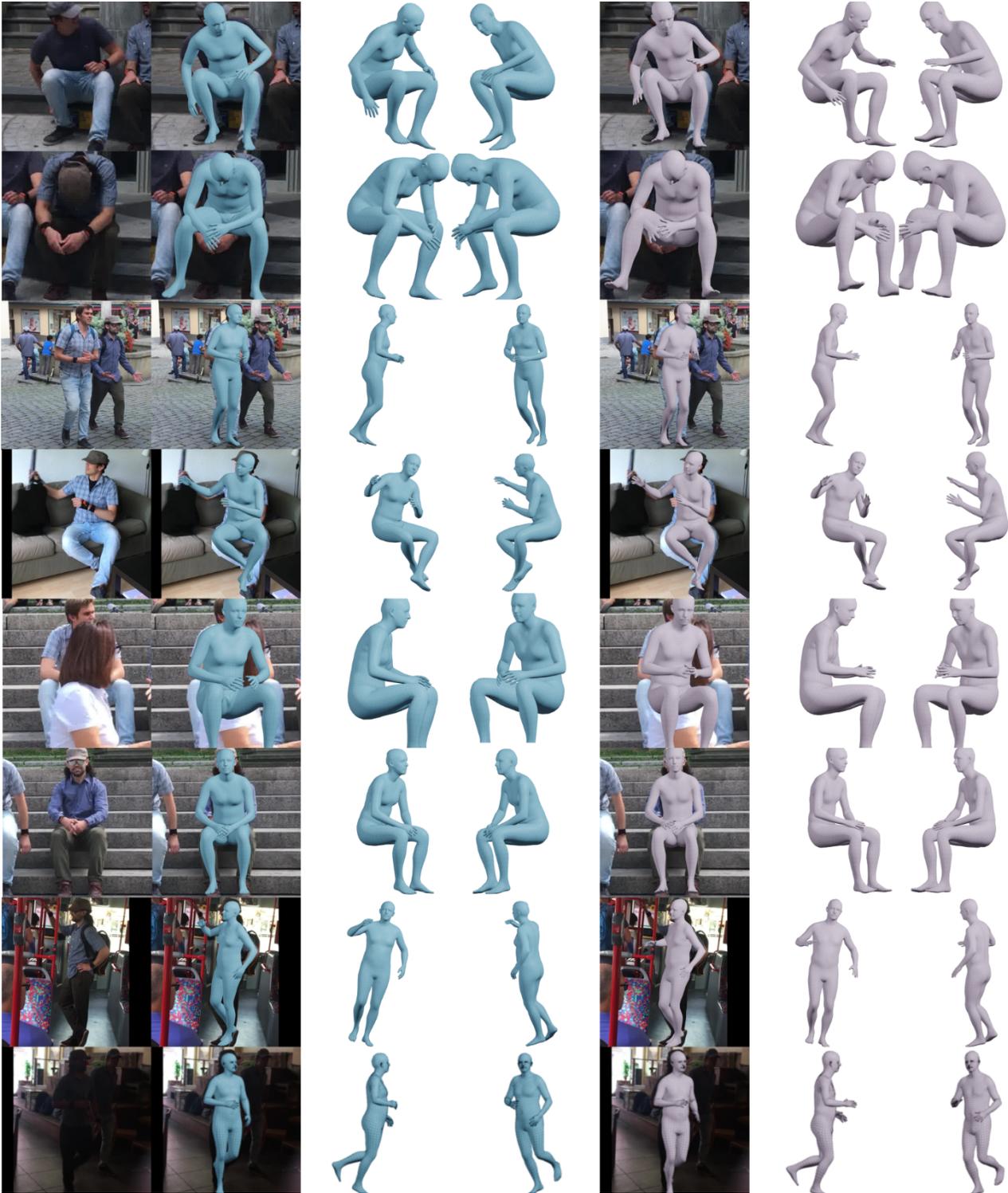


Figure 11. Qualitative results on the self-contact subset of 3DPW. We find all images with an improvement on MPJPE and PA-MPJPE  $\geq 10$  mm. From this subset, we select interesting poses. Left column, RGB image for reference. In blue, TUCH result and in violet, the SPIN result.

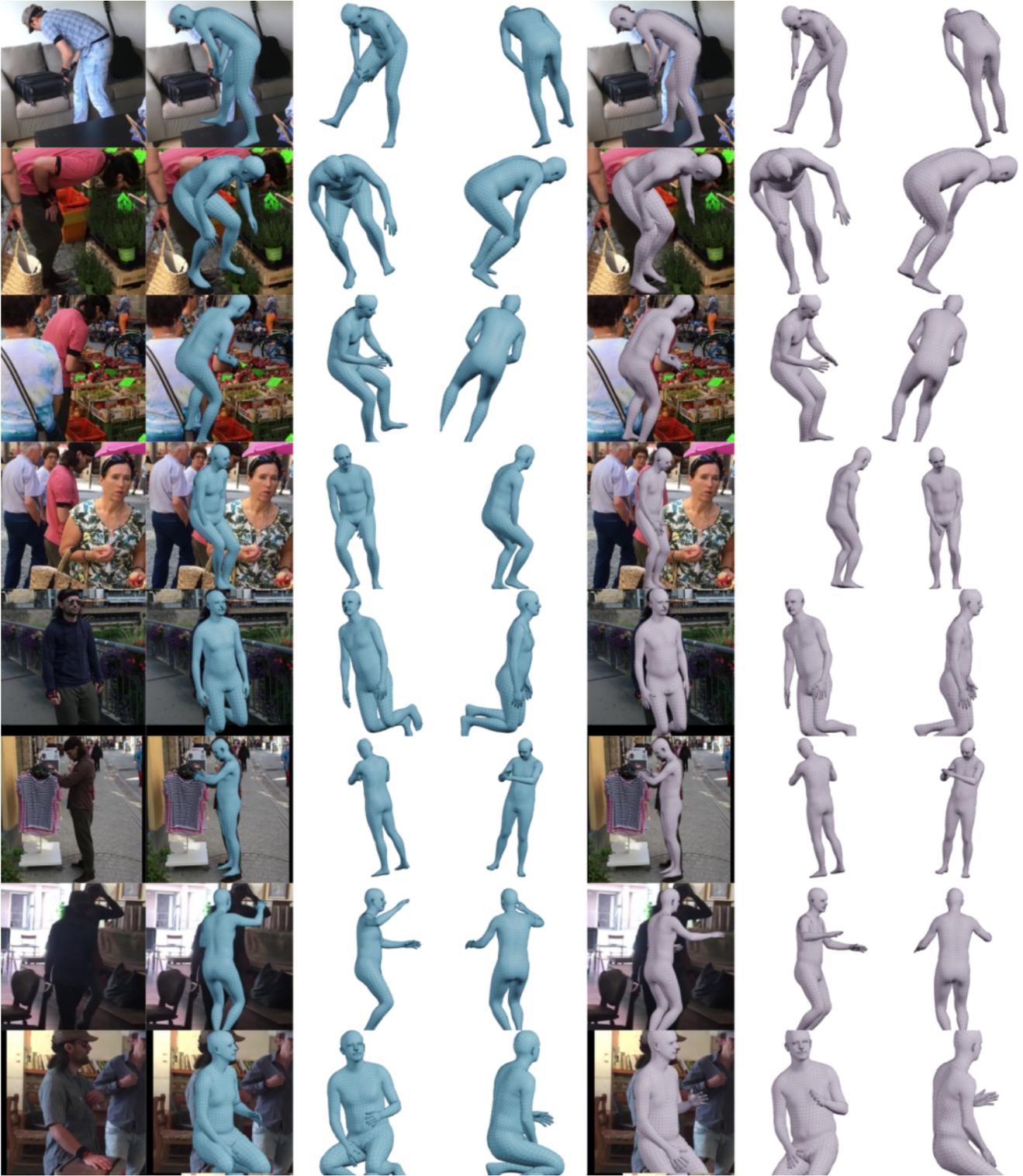


Figure 12. Qualitative results on the self-contact subset of 3DPW. We find all images where SPIN is better than TUCH by at least 10 mm for MPJPE and PA-MPJPE. From this subset, we select interesting poses. Left column, RGB image for reference. In blue, TUCH result and in violet, the SPIN result.

## References

- [1] Alec Jacobson, Ladislav Kavan, and Olga Sorkine-Hornung. Robust inside-outside segmentation using generalized winding numbers. *ACM Trans. Gr.*, 32(4):1–12, 2013. 1
- [2] Hanbyul Joo, Natalia Neverova, and Andrea Vedaldi. Exemplar fine-tuning for 3D human pose fitting towards in-the-wild 3D human pose estimation. *arXiv:2004.03686*, 2020. 5, 6
- [3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 5
- [4] Nikos Kolotouros, Georgios Pavlakos, Michael J. Black, and Kostas Daniilidis. Learning to reconstruct 3D human pose and shape via model-fitting in the loop. pages 2252–2261, 2019. 5
- [5] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989. 4
- [6] Priyanka Patel, Chun-Hao Paul Huang, Joachim Tesch, David Hoffmann, Shashank Tripathi, and Michael J Black. AGORA: Avatars in geography optimized for regression analysis. 2021. 4