LatentHuman: Shape-and-Pose Disentangled Latent Representation for Human Bodies Supplementary Material

Sandro Lombardi[1*] Bangbang Yang[2*] Tianxing Fan[2]
Hujun Bao[2] Guofeng Zhang[2] Marc Pollefeys[1,3] Zhaopeng Cui[2†]
[1]ETH Zurich [2]State Key Lab of CAD&CG, Zhejiang University [3]Microsoft

In this supplementary material, we provide more details on data preparation (Sec. A.1), the used training losses (Sec. A.2), and the progressive training scheme (Sec. A.3). The architecture and the training procedure for our VJointer module are described in Sec. B.1 and Sec. B.2 respectively. Finally, we provide more representation comparison (Sec. C.1) and pose tracking results (Sec. C.2), and introduce the experiment of shape swapping and pose animation in Sec. C.3.

A. Training Details

A.1. Data Preparation

As mentioned in Sec. 3.3, we sample $M = 4000$ surface samples directly from posed SMPL meshes. We use the associated skinning weights to assign each vertex and face of the ground truth mesh a unique label corresponding to the joint with the highest skinning weight. This allows us to draw more surface samples from body parts containing fine details like facial features and fingers at hands. Specifically speaking, we use the sampling weights shown in Table A. We use this part-based sampling strategy in all our experiments except for the pose tracking experiment where we perform a uniform sampling of points.

In order to generate $M = 4000$ near-surface samples, we compute a random offset with a standard deviation of $\sigma = 0.01$ and add it to each sampled surface point. For the computation of $Q = 800$ random off-surface points, we first compute tight bounding boxes for each body part, determined by the labeled and sampled surface points. We then randomly sample points within those bounding boxes. In practice we enlarge the part bounding boxes by a factor of 2 before sampling.

For a fair comparison, we also train NASA and LEAP with the same SMPL model. Even though LEAP relies on the SMPL-H model we didn't observe a noticeable performance difference as LEAP's reconstructed hands usually do not show significant articulation.

Body Parts	Joint Nr.	Weights
Body (Pelvis, Spine 1-3, L. & R. Collar)	0,3,6,9,13,14	15% (2.5%)
Legs (L. & R. Hip, L. & R. Knee)	1,2,4,5	10% (2.5%)
Feet (L. & R. Ankle, L. & R. Foot)	7,8,10,11	20% (5%)
Arms (L. & R. Shoulder, L. & R. Elbow)	16,17,18,19	10% (2.5%)
Hands (L. & R. Wrist, L. & R. Hand)	20,21,22,23	30% (7.5%)
Head (Neck, Head)	12,15	15% (5% + 10%)

Table A: **Sampling weights.** During training, we increase the weights of sampled query points for body parts which require high-frequency details, *e.g.*, head, hands and feet, while decreasing the weight for others, *e.g.*, the body, arms and legs.

Data Preparation for Model Fitting Experiment. For the model fitting experiment of Sec. 4.4, we create a dataset where SMPL pose parameters are taken from AMASS [6] DFaust subsets while SMPL shape parameters are provided by the SURREAL [11] dataset. Those parameters are then combined and the SMPL model is used to generate ground truth meshes. As mentioned in Sec. 4.1, the training split consists of 10 subjects and uses all except 1 randomly withhold sequence per subject. For the shape parameters, we use 50 randomly selected male and female shapes respectively.

Data Preparation for Fine-tuning Experiment. In our experiment of Sec. 4.6, we fine-tune our model on raw DFaust [1] body scans and scans from the CAPE [5, 9] clothed human dataset. Since the scanned meshes from the CAPE dataset tend to contain large clutter, especially on the ground, we follow Saito *et al*. [10] and manually remove all mesh faces and associated vertices below a preset ground plane, located at $y_{\text{ground}} = -0.562$. As the mesh scans from the CAPE and the DFaust dataset are by default not aligned with the coordinate system used by the SMPL model, we need to transform them in order to maintain the same data processing steps as in the other experiment setups. To this end, both datasets provide transformations which map and align each respective scan to the SMPL-based coordinate system. After applying those transformations, we follow

the same steps as outlined in the beginning of this section for obtaining query locations.

For our supplementary video, we follow Saito *et al*. [10] and replace the feet of CAPE scans with the surface from the SMPL mesh since the accessible training poses are severely limited and don't include poses which lift the feet, *e.g*., walking and jumping motions.

A.2. Losses

As mentioned in Sec. 3.3, we rely on several loss functions with different weights. We add the loss functions mentioned in Eq. 4 to both, the part-specific SDF predictions and the combined overall SDF predictions. Specifically, we use $\lambda_{\text{m}} = 70$, $\lambda_{\text{nm}} = 35$, $\lambda_{\text{n}} = 7$, $\lambda_{\text{e}} = 17.5$ for overall losses and $\lambda_{\text{dual-m}} = 30$, $\lambda_{\text{nm}} = 15$, $\lambda_{\text{n}} = 3$, $\lambda_{\text{e}} = 0$ for part-specific losses, *i.e*., in practice we omit the eikonal loss for part-specific outputs. For the part-specific one-sided non-manifold loss, we use $\lambda_{\text{osnm}} = 15$ while for the latent loss, a weight of $\lambda_{\text{zs}} = 0.001$ is applied. Note that for two-sided non-manifold losses, we use a hyper parameter of $\alpha = 5$ while for the one-sided non-manifold loss, we use a truncation distance of $\delta = 0.01$.

A.3. Progressive training

We adopt the progressive training scheme (a.k.a annealed positional encoding) from Park *et al*. [7], and rewrite the positional encoding $\gamma_\alpha(\mathbf{x})$ as:

$$\gamma_\alpha(\mathbf{x}) = \left(\mathbf{x}, \cdots, w_k(\alpha)\sin\left(2^k\pi\mathbf{x}\right), w_k(\alpha)\cos\left(2^k\pi\mathbf{x}\right), \cdots\right), \tag{1}$$

$$w_k(\alpha) = \frac{(1-\cos(\pi\cdot\text{clamp}(\alpha-k,0,1))}{2}, \tag{2}$$

$$\alpha(t) = t\frac{K}{P} \quad \text{with} \quad t\in[0,1], \tag{3}$$

where $k \in \{1, \cdots, K\}$ with $K = 10$ denotes the k-th frequency order of the positional encoding. $w_k(\alpha)$ is the annealed weight for the k-th order item in $\gamma_\alpha(\mathbf{x})$ and $\alpha(t)$ is the annealing hyper parameter, determining the speed at which the network switches from a low-frequency bias to a high-frequency bias. t is the training progress, *i.e*., current number of iteration divided by the total number of iterations. Finally, $P = 0.2$ controls when α reaches the maximum number of frequencies, *i.e*., the network's largest bias towards high frequencies is reached after the training progress achieves 20%.

B. Details of VJointer

As mentioned in Sec. 3.1, we use a variational autoencoder (VAE) [4] named VJointer to learn a human skeleton joint prior.

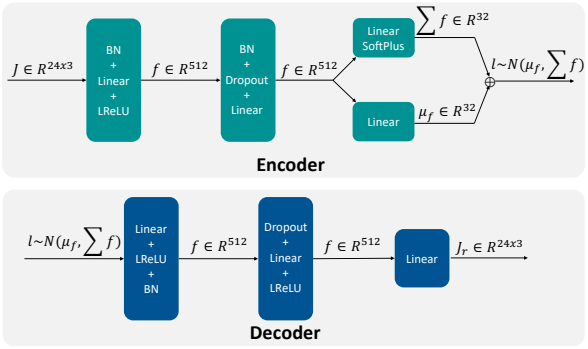

Figure A: **Architecture of VJointer.** $J \in R^{24\times3}$ denotes the input canonical skeleton joints, $J_r \in R^{24\times3}$ represents the reconstructed canonical skeleton joints, and BN denotes the batch normalization.

B.1. Model Architecture

Inspired by VPoser [8], we design the architecture of VJointer as shown in Fig. A, where we set the hidden dimension of linear layers to 512, use a LeakyReLU activation with negative slope 0.1 and apply a Dropout layer with probability 0.1.

B.2. Training

We train VJointer with 3830 human bodies from the SURREAL dataset [11], as it possesses better shape variety than the AMASS dataset. Specifically, we construct the SMPL body models with the shape parameters of the dataset, and compute the canonical skeleton joints with the SMPL joint regressor, which yields 24 joints for each human body. We then flatten these joints to obtain a 72-dimensional vector, which will be served as the training input of VJointer. During the training process, we use a L1-loss for the joint regression error $\mathcal{L}_{loc}$ and a KL-divergence loss $\mathcal{L}_{KL}$ to supervise the model. The total loss function $\mathcal{L}_{joint}$ can be formulated as:

$$\mathcal{L}_{joint} = \mathcal{L}_{KL} + \lambda_{loc}\mathcal{L}_{loc}, \tag{4}$$

$$\mathcal{L}_{KL} = KL(d(L|J)||\mathcal{N}(0,I)), \tag{5}$$

$$\mathcal{L}_{loc} = |J - J_r|, \tag{6}$$

where $L \in R^{32}$ denotes the latent code of the autoencoder and $\lambda_{loc} = 400$ determines the weighting of the individual loss terms. $J \in R^{24\times3}$ and $J_r \in R^{24\times3}$ represent the ground-truth and the estimated canonical skeleton joints respectively. We use the Adam optimizer [3] with a learning rate of 0.0001.

C. More Experiment Results

In addition to the results presented in the paper, we provide more qualitative results and one additional experiment.

C.1. Representation Comparison

Fig. B and Fig. C provide additional qualitative reconstruction results for the AMASS / DFaust and the AMASS / MoVi dataset. In contrast to ours, the reconstructed results from NASA [2] miss high-frequency details like facial expressions and fingers highlighted by the dashed rectangles. In the case of LEAP, the results show comparable levels of detail, but LEAP introduces visual artifacts or contains missing surface areas. The same experiment setup as explained in Sec. 4.2 was used.

C.2. Pose Tracking

We provide more pose tracking and retargeting results in Fig. D. The experimental setup from Sec. 4.5 was used. We can see that our pose tracking works well and the recovered poses can be retargeted to cartoon characters as our model is designed based on the kinematic model.

C.3. Shape Swapping and Pose Animation

In this section, we introduce how we conduct the shape swapping and pose animation in Fig. 1 of our main paper, and demonstrate the effectiveness of LatentHuman in learning shape-and-pose disentangled representation. Specifically speaking, we take the shape and pose representations of one subject and replace one of them with the corresponding representation of another subject while keeping the other fixed. In Fig. E, the first two rows show the result of swapping the shape of one subject with 3 other subjects, visualized in different poses. Analogously, the last two rows show the result of animating one subject with the poses taken from another subject. One can observe that the poses are not influenced by an exchanged shape code and vice versa which indicates that the two spaces are well disentangled. Please refer to the supplementary video for a vivid animation of this experiment.

C.4. Model fitting with different number of points.

We visualize the model fitting results with different number of points in Fig. F. Even with only 500 points, the shape and pose can be successfully recovered, demonstrating the efficacy of the proposed method.

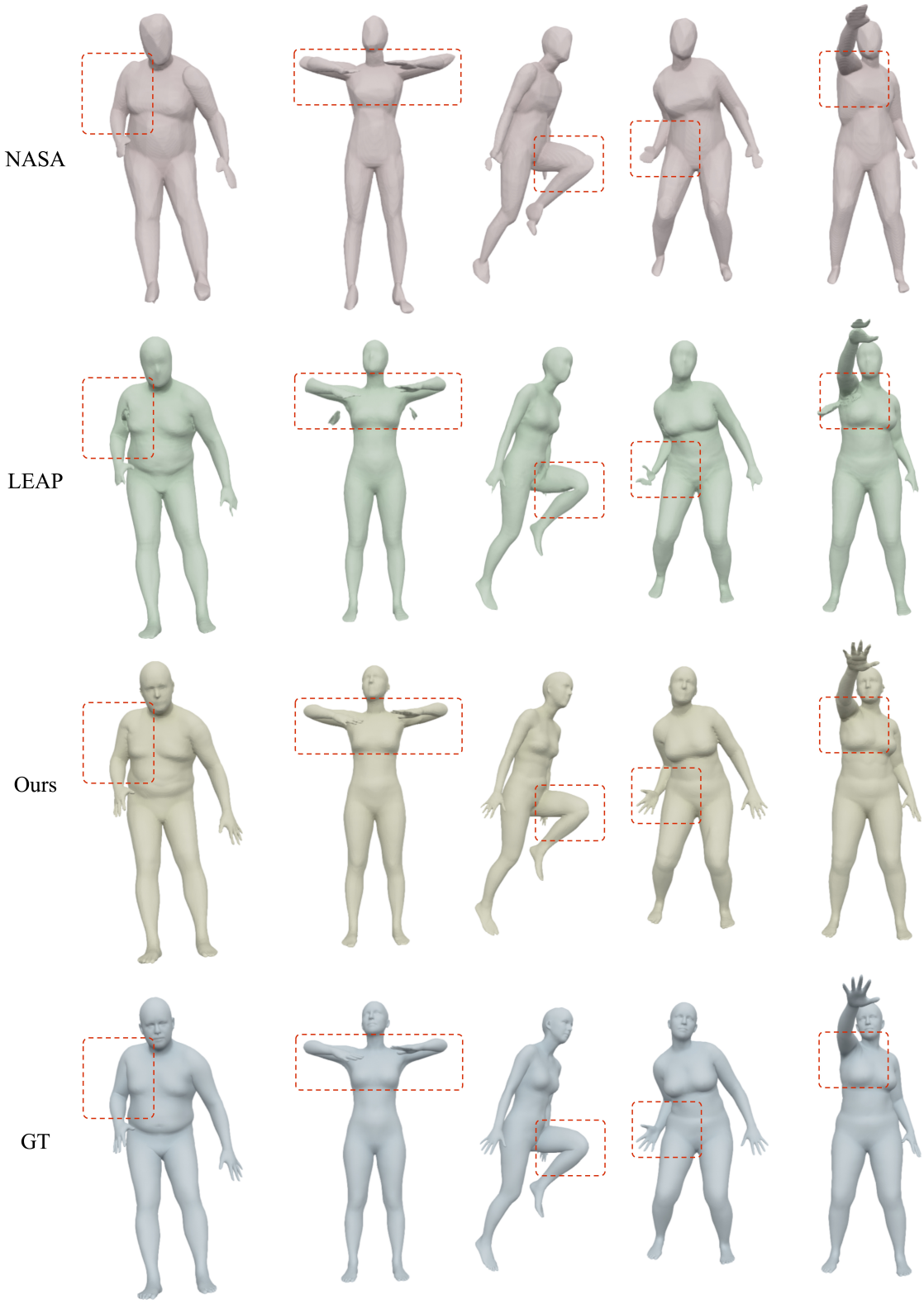

Figure B: **More qualitative comparisons on AMASS / DFaust.**

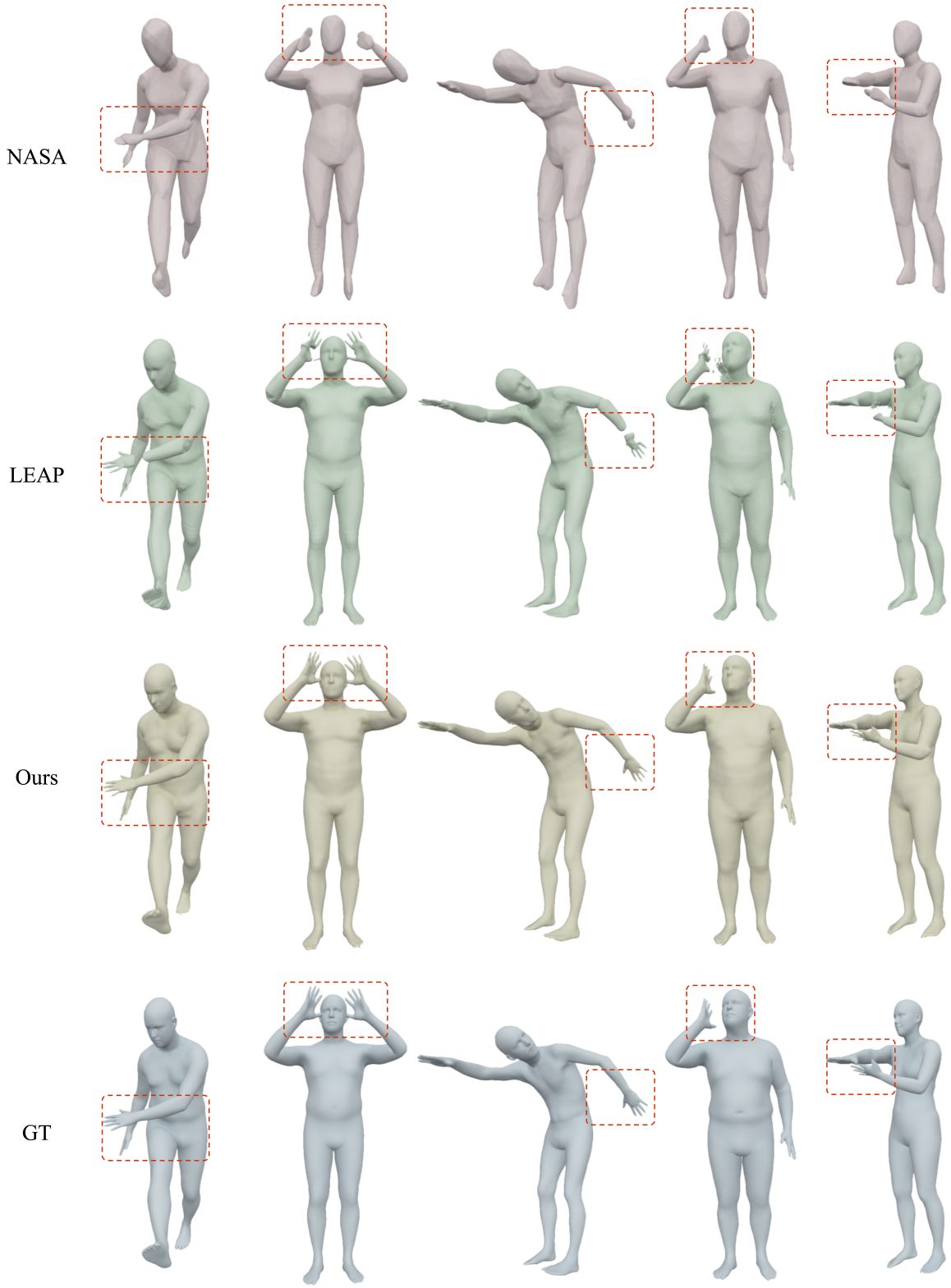

Figure C: **More qualitative comparisons on AMASS / MoVi.**

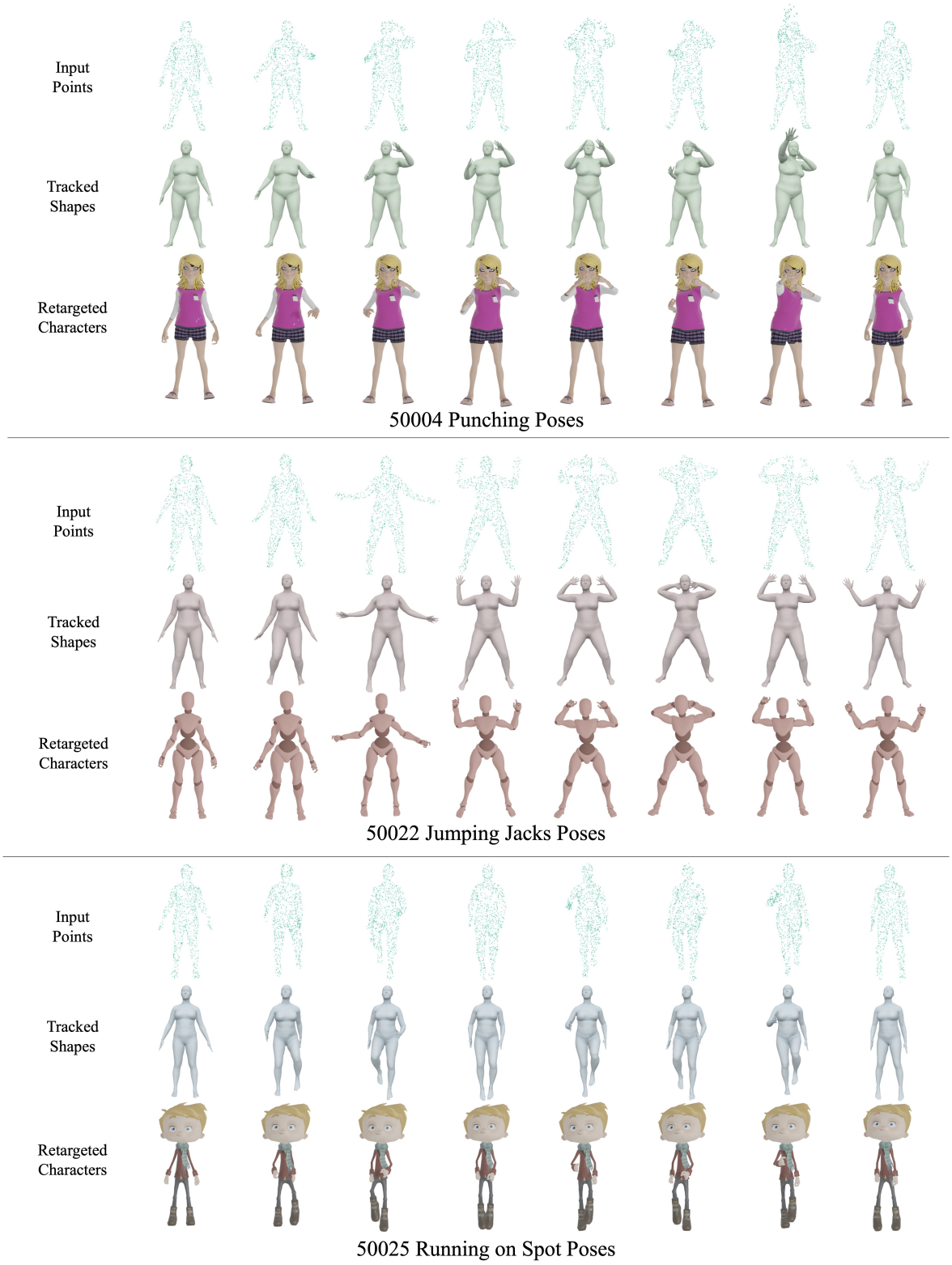

Figure D: **More pose tracking results.** The first row of each example shows the tracked input point cloud. The reconstructed tracked shapes are shown in the 2nd row. Finally the 3rd row shows the cartoon characters animated with the tracked pose.

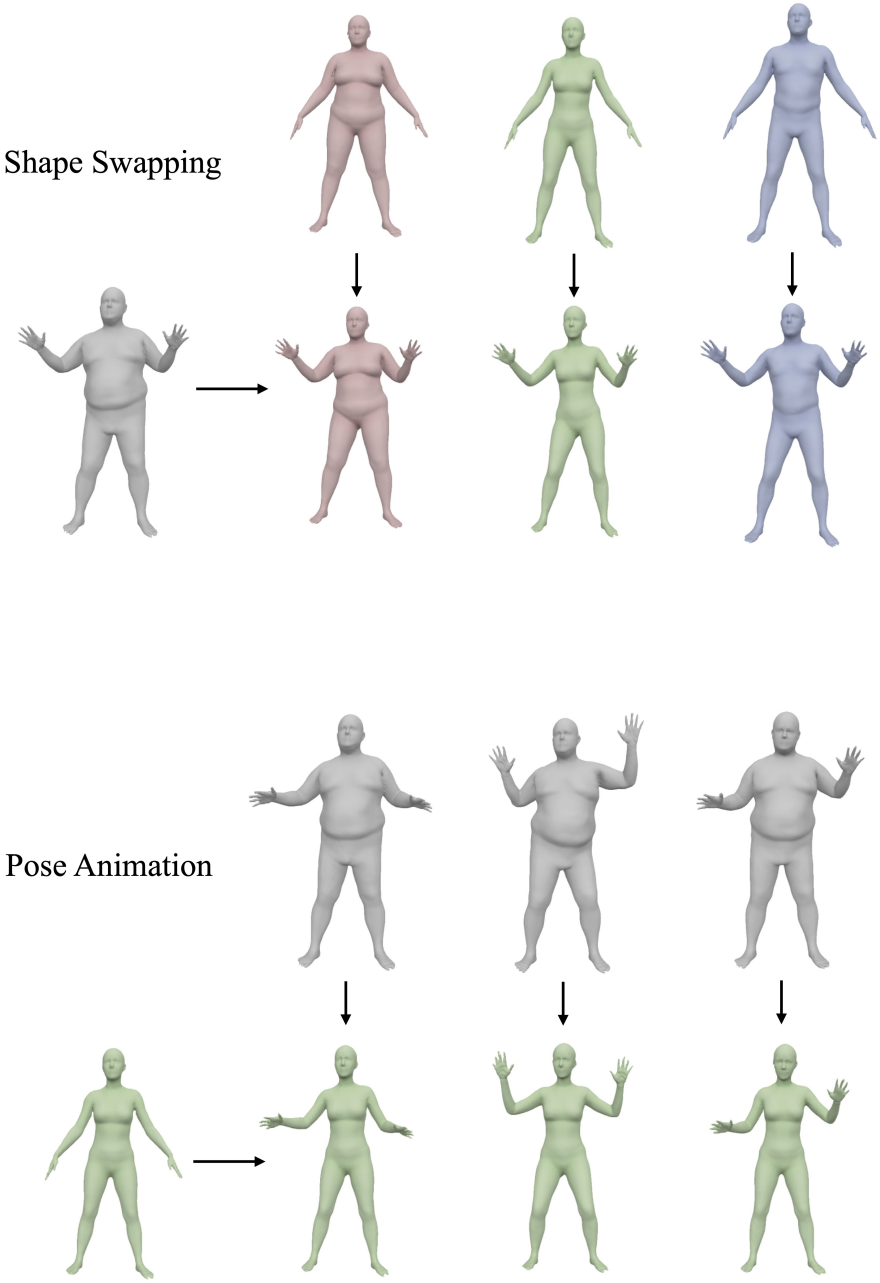

Figure E: **Shape swapping and pose animation.** The first two rows show the shape swapping of one subject (gray) with 3 other subjects (red, green and blue), and the last two rows show the animation of one subject (green) with the poses from another subject (gray).

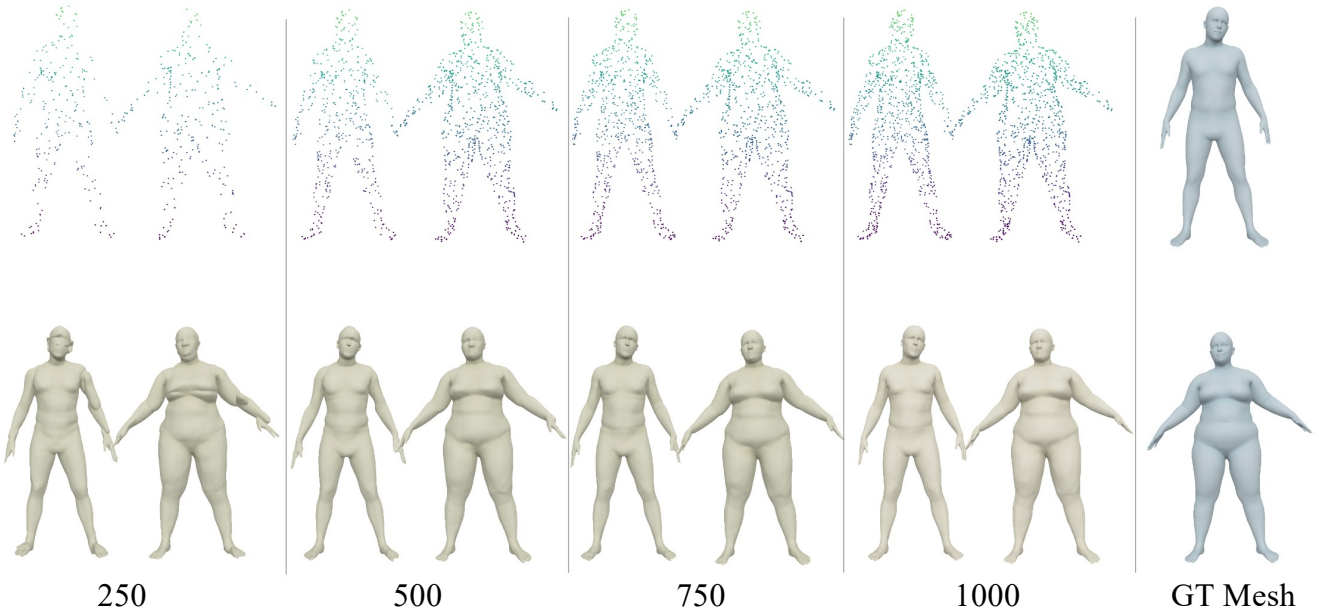

Figure F: **Model fitting with different number of points.** We fit our representation to 250 $\sim$ 1000 input points (top row in the left four columns) of human bodies in a casual A-pose by jointly optimizing the shape and pose. The reconstructed human bodies are visualized in yellow meshes (bottom row in the left four columns), and the ground-truth human bodies are visualized in blue meshes (right column).

References

[1] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: registering human bodies in motion. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5573–5582. IEEE Computer Society, 2017. 1

[2] Boyang Deng, J. P. Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. NASA Neural Articulated Shape Approximation. In *Computer Vision – ECCV 2020*, 2020. 3

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 2

[4] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 2

[5] Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3d people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478, 2020. 1

[6] 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. 1

[7] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Deformable neural radiance fields. *CoRR*, abs/2011.12948, 2020. 2

[8] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 2

[9] Gerard Pons-Moll, Sergi Pujades, Sonny Hu, and Michael J. Black. Clothcap: seamless 4d clothing capture and retargeting. *ACM Trans. Graph.*, 36(4):73:1–73:15, 2017. 1

[10] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J. Black. SCANimate: Weakly Supervised Learning of Skinned Clothed Avatar Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2886–2897, 2021. 1, 2

[11] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 109–117, 2017. 1, 2