Supplementary Material for SCANimate: Weakly Supervised Learning of Skinned Clothed Avatar Networks

Shunsuke Saito Jinlong Yang Qianli Ma Michael Black
Max Planck Institute for Intelligent Systems, Tübingen, Germany

1. Implementation details

1.1. Network Architectures

Our forward and inverse skinning networks are based on multi-layer perceptrons, where the intermediate neuron size is $(256, 256, 256, 24)$ with a skip connection from the input feature to the 2nd layer, and nonlinear activations using LeakyReLU except the last layer that uses softmax to obtain normalized skinning weights. As an input, we take the Cartesian coordinates of a queried location, which is encoded into a high dimensional feature using the positional encoding [6] with up to 6-th and 8-th order Fourier features for the forward skinning net $g^c_{\Theta_1}(\cdot)$ and the inverse skinning net $g^s_{\Theta_2}(\cdot)$, respectively. Note that the inverse skinning net $g^s_{\Theta_2}(\cdot)$ takes a latent embedding $\boldsymbol{z}^s_i \in \mathbb{R}^{64}$ as an additional input in order to learn the skinning weights of scans in different poses.

To model the geometry of clothed humans in a canonical pose, we also use a multi-layer perceptron $f_\Phi(\cdot)$, where the intermediate neuron size is $(512, 512, 512, 343, 512, 512, 1)$ with a skip connection from the input feature to the 4th layer, and nonlinear activations using softplus with $\beta = 100$ except the last layer as in [2]. The input feature consists of the Cartesian coordinates of a queried location, which are encoded using the positional encoding of up to 8-th order Fourier features, and the localized pose encoding in $\mathbb{R}^{92}$. The texture inference network uses the same architecture as the geometry module $f_\Phi(\cdot)$ except the last layer with 3 dimensional neurons for color prediction, and the input layer replaced with the concatenation of the same input and the second last layer of $f_\Phi(\cdot)$ so that the color module is aware of the underlying geometry.

1.2. Training Procedure

Our training consists of three stages. First, we pretrain $g^c_{\Theta_1}(\cdot)$ and $g^s_{\Theta_2}(\cdot)$ with the following relative weights: $\lambda_B = 10.0$, $\lambda_S = 1.0$, $\lambda_{C'} = 0.0$, $\lambda_{C''} = 0.0$, $\lambda_{Sp} = 0.001$, $\lambda_{Sm} = 0.0$, and $\lambda_Z = 0.01$. After pretraining, we jointly train $g^c_{\Theta_1}(\cdot)$ and $g^s_{\Theta_2}(\cdot)$ using the proposed cycle consistency constraint with the following weights: $\lambda_B = 10.0$, $\lambda_S = 1.0$, $\lambda_{C'} = 1.0$, $\lambda_{C''} = 1.0$, $\lambda_{Sp} = 0.001$, $\lambda_{Sm} = 0.1$, and $\lambda_Z = 0.01$. We multiply $\lambda_{C''}$ by 10 for the second half of the training iterations. For the two stages above, we use 6890 points of the SMPL vertices and 8000 points uniformly sampled on the scan data, which is dynamically updated at every iteration.

Once the training of the skinning networks is complete, we fix the weights of $g^c_{\Theta_1}(\cdot)$, $g^s_{\Theta_2}(\cdot)$, and $\{\boldsymbol{z}^s_i\}$, and train the geometry module $f_\Phi(\cdot)$ with the following hyper parameters: $\lambda_{igr} = 1.0$, $\lambda_o = 0.1$, and $\alpha = 100$. To compute E_{LS}, we uniformly sample 5000 points on the scan surface at each iteration. We compute E_{IGR} by combining 2000 points within a bounding box and 10000 points perturbed with the standard deviation of 10cm from the surface geometry, half of which is sampled from the scans and the remaining from the SMPL body vertices. Note that E_O uses only 2000 points sampled from the bounding box to avoid overly penalizing zero crossing near the surface.

We train each stage with the Adam optimizer with learning rates of 0.004, 0.001, and 0.001, respectively. They are decayed by the factor of 0.1 at $1/2$ and $3/4$ of the training iterations. The first stage runs for 80 epochs and the second for 200 epochs.

1.3. Texture Inference

To model texture on the implicit surface, we model texture fields parameterized by a neural network, denoted as $f^c(\boldsymbol{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, following [7, 9]. Given the ground-truth color $c(\boldsymbol{x})$ at a location x on the surface, we learn the network weights of $f^c(\cdot)$ by minimizing the $L1$ reconstruction loss: $|f^c(\boldsymbol{x}) - c(\boldsymbol{x})|$. We sample 5000 points from the input scans at every iteration and optimize using the Adam optimizer with a learning rate of 0.001 and the same decay schedule as the geometry module. We train the texture module for $1.8M$ iterations.

1.4. Other Details

Concave region detection. We exclude concave regions from the smoothness constraint to avoid propagating incorrect skinning weights at the self-intersection regions. We detect them by computing the mean curvature on the surface of scans with the threshold of 0.2. Note that while we

empirically find our detection algorithm is sufficient for our training data, utilizing external information such as body part labels is possible when available to improve robustness.

Obtaining canonical body. The canonicalized body $\mathbf{B}_i^c$ in Eq. (5) is a body model of the subject in a canonical pose with pose dependent deformations. We obtain the pose correctives by activating pose-aware blend shapes in the SMPL model [3] given the body pose θ at frame i.

Removing distorted triangles. When the input scans are canonicalized, triangle edges that belong to self-intersection regions are highly distorted. As these regions must be separated in the canonical pose, we remove all triangles for which any edge length is larger than its initial length multiplied by 4.

2. Discussion

2.1. Latent Autodecoding

The purpose of learning $g^s(\cdot, \boldsymbol{z})$ is to stably canonicalize raw scans. To this end, we use auto-decoding $\boldsymbol{z}$ as in [8] for the following advantages. Auto-decoding self-discovers the latent embedding $\boldsymbol{z}$ such that the loss function is minimized, allowing the network to better distinguish each scan regardless of the similarity in the pose parameters. Thus, $\boldsymbol{z}$ can implicitly encode not only pose information but also anything necessary to distinguish each frame. Furthermore, due to no dependency on pose parameters, auto-decoding is more robust to the fitting error of the underlying body model. As a baseline we replace autodecoding by regressing skinning weights on pose parameters of a fitted SMPL body. We use the energy function E_{cano} in Eq. (4) without the term of E_Z to evaluate the performance of the two. While pose regression results in 0.043, autodecoding achieves a much lower local minimum at 0.025, showing superior performance against the baseline.

2.2. Combining Skinning Networks

As in Eq. (2) in the main paper, g^c and g^s are formulated separately. This is in accordance with the idea of predicting skinning weights for both forward and backward transformations. However, if one considers the skinning networks in another point of view, particularly when regarding them as mappings from 3D space coordinates conditioned on different frames to skinning weights, it is clear that g^c is a special case of g^s. Thus in practical implementation, one can either set up two networks corresponding to g^c and g^s, respectively, or set up a single networks in an autodecoder manner with a single common latent vector $\boldsymbol{z}^c$ for all the forward skinning weights prediction and per-frame latent vectors $\boldsymbol{z}_i^s$ for inverse skinning weights prediction in each posed frame.

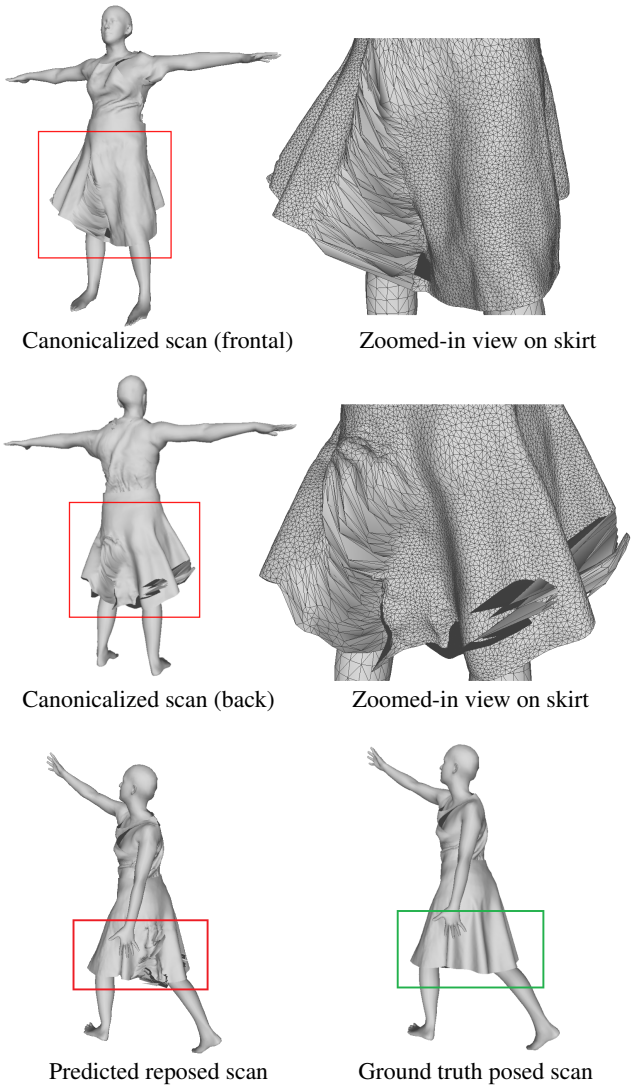
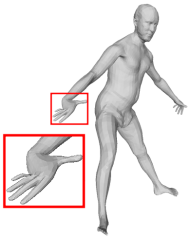


Figure 1: Failure cases of canonicalizing a clothed human with a synthetic skirt. We show surface triangles in the zoomed-in images to highlight the severe stretching artifacts of the skirt between legs.

2.3. Failure cases.

As mentioned in the main paper, while the current pipeline performs well for clothing that is topologically similar to the body, the method may fail for clothing, like skirts, whose topology may deviate significantly. Fig. 1 shows a failure case of canonicalizing a person with a skirt synthetically generated using a physics-based simulation. The SMPL-guided initialization of skinning weights fails recovering from poor local minima. We leave for future work a garment-specific tuning of hyperparameters and more robust training schemes for various clothing types.

2.4. CAPE Dataset Limitation.



Some frames of the CAPE dataset [4] contain erroneous body fitting around the wrists and ankles, as shown in the right inset figure, resulting in unnecessary distortions around the regions. Due to the smoothness regularization in our method, such a distortion can be propagated to the nearby regions, and hence a larger region may be discarded. However, the proposed shape learning method complements such a missing region from other canonicalized scans, and our reconstructed Scanimats do not suffer from the small errors in pose fitting.

3. Additional Qualitative Results

Locally Pose-aware Shape Learning. Fig. 2, an extended Figure of Fig. 5 in the main paper, shows more qualitative comparison on pose encoding with different sizes of training data.

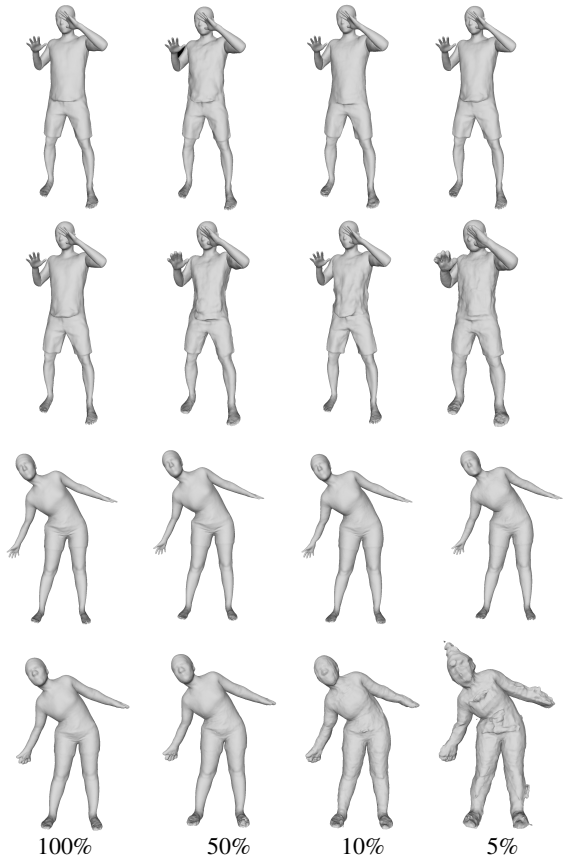


Figure 2: **Evaluation of pose encoding with different sizes of training data.** Top row: our local pose encoding. Bottom row: global pose encoding. While the global pose encoding suffers from severe overfitting artifacts, our local pose encoding generalizes well even if data size is severely limited.

Comparison with the SoTA methods. Fig. 3, an extended Figure of Fig. 6 shows more qualitative comparison with the SoTA methods.

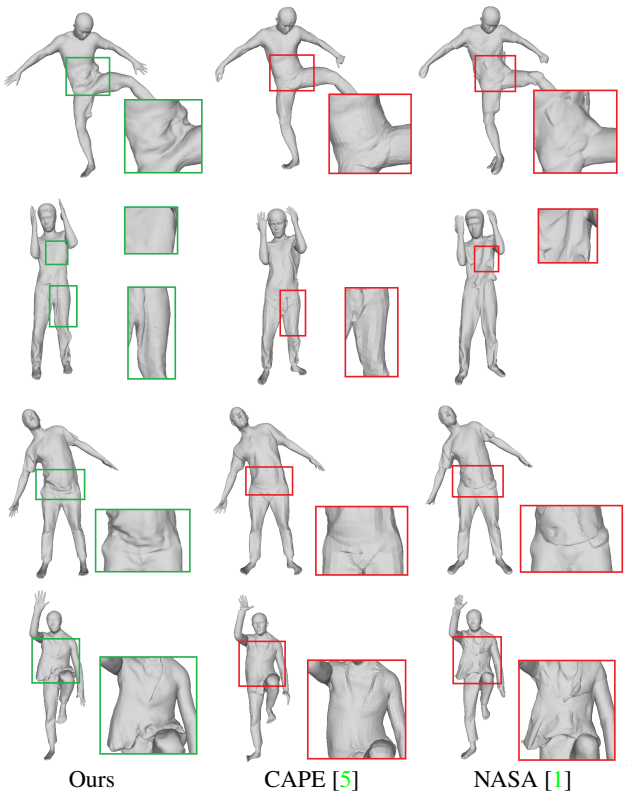


Figure 3: **Comparison with the SoTA methods.** We show qualitative results on the extrapolation task, illustrating the advantages of our method as well as the limitations of the existing approaches.

Textured Scanimats Fig. 4, an extended Figure of Fig. 7, shows more examples of textured Scanimats.

Figure 4: **Textured Scanimats.** Our method can be extended to texture modeling, allowing us to automatically build a Scanimat with high-resolution realistic texture.

Please watch the video at https://scanimate.is.tue.mpg.de for animated results.

References

[1] Boyang Deng, John P. Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey E. Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. NASA neural articulated shape approximation. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VII*, volume 12352 of *Lecture Notes in Computer Science*, pages 612–628. Springer, 2020. 3

[2] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3789–3799. PMLR, 2020. 1

[3] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015. 2

[4] 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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6468–6477. IEEE, 2020. 3

[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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 6468–6477. IEEE, 2020. 3

[6] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 405–421. Springer, 2020. 1

[7] Michael Oechsle, Lars M. Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 4530–4539. IEEE, 2019. 1

[8] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 165–174. Computer Vision Foundation / IEEE, 2019. 2

[9] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Hao Li, and Angjoo Kanazawa. PIFu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2304–2314. IEEE, 2019. 1