Supplementary Materials for SNARF: Differentiable Forward Skinning for Animating Non-Rigid Neural Implicit Shapes

Xu Chen[1,3] Yufeng Zheng[1,3] Michael J. Black[3] Otmar Hilliges[1] Andreas Geiger[2,3]
[1]ETH Zürich, Department of Computer Science [2]University of Tübingen
[3]Max Planck Institute for Intelligent Systems, Tübingen

Abstract

*In this **supplementary document**, we first provide implementation details of our method in Sec. 1. We then provide details regarding the evaluation protocol and the implementation of baseline methods in Sec. 2. Finally, we show additional qualitative results in Sec. 3.*

1. Implementation Details

In this section, we provide technical details of our method.

1.1. Architecture

We implemented our models in PyTorch [12]. Our architectures for the occupancy network and the skinning network are illustrated in Fig. 1. Note that we chose a smaller network size for the skinning weight network, as skinning weights are typically smooth and thus don't require a high-capacity network to be modeled well. We use geometric initialization [2] for the occupancy network's weights and PyTorch's default initialization for the skinning network weights. The pose condition $\mathbf{p}$ for the canonical occupancy network is obtained by concatenating all axis angles. No positional encoding [11] is used in the experiments on 3D minimally clothed humans to enable a fair comparison to the NASA baseline. For our results on 3D clothed humans, we apply positional encoding [11] with 4 frequency components on the input points to better model local details, *e.g.*, wrinkles.

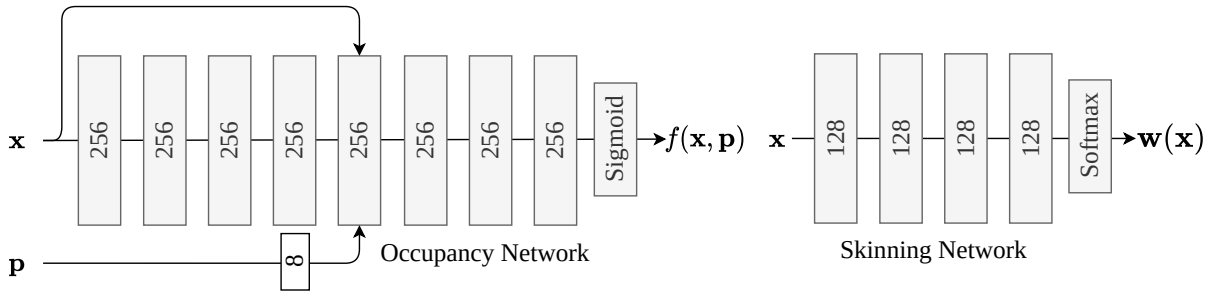


Figure 1: **Network Architectures.** Each block represents a linear layer with its output dimension specified in the inset, followed by a weight normalization layer [13] and a softplus [6] activation layer. An exception is the block at the bottom left which represents a linear layer for reducing the dimension of the pose condition to 8.

1.2. Root Finding

We use Broyden's method [4] for our correspondence search. To find the solution to the equation $\mathbf{g}(\mathbf{x}) = \mathbf{0}$, Broyden's method iteratively updates the solution estimate $\mathbf{x}^k$ via

$$\Delta \mathbf{x}^k = \mathbf{x}^k - \mathbf{x}^{k-1} \tag{1}$$

$$\Delta \mathbf{g}^k = \mathbf{g}(\mathbf{x}^k) - \mathbf{g}(\mathbf{x}^{k-1}) \tag{2}$$

$$(\mathbf{J}^k)^{-1} = (\mathbf{J}^{k-1})^{-1} + \frac{\Delta \mathbf{x}^k - (\mathbf{J}^{k-1})^{-1}\Delta \mathbf{g}^k}{(\Delta \mathbf{x}^k)^T \cdot (\mathbf{J}^{k-1})^{-1} \cdot \Delta \mathbf{g}^k}(\Delta \mathbf{x}^k)^T(\mathbf{J}^{k-1})^{-1} \tag{3}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\mathbf{J}^k)^{-1} \cdot \mathbf{g}(\mathbf{x}^k) \tag{4}$$

where $\mathbf{J}^k$ is the approximation of the Jacobian matrix of $\mathbf{g}$. In our case, we have $\mathbf{g}(\mathbf{x}) = \mathbf{d}_{\sigma_w}(\mathbf{x}^k, \boldsymbol{B}) - \mathbf{x}'$. In our experiments, we set the maximum number of update steps to 50 and the convergence threshold to 10^{-5}. After each update, points with errors lower than the convergence threshold are considered converged and excluded in further updates.

1.3. Losses

As described in the main paper, our main training loss is the binary cross entropy loss $\mathcal{L}_{BCE}(o(\mathbf{x}', \mathbf{p}), o_{gt}(\mathbf{x}'))$ between the predicted occupancy of the deformed points $o(\mathbf{x}', \mathbf{p})$ and the corresponding ground-truth occupancy $o_{gt}(\mathbf{x}')$ for all posed 3D meshes of a single subject that we use as observations. For complex shapes and articulation (*e.g.*, 3D humans), we add a small auxiliary loss to guide learning in early iterations. Towards this goal, we randomly sample points $\mathbf{x}_{bone}$ along the bones connecting joints in canonical space and encourage their occupancy probabilities to be one, by minimizing a binary cross entropy loss $\mathcal{L}_{bone}$. Moreover, we encourage the skinning weights of all joints $\mathbf{x}_{joint}$ in canonical space to be equal to 0.5 for the respective neighboring bones

$$\mathcal{L}_{bone} = \mathcal{L}_{BCE}(f_{\sigma_f}(\mathbf{x}_{bone}, \mathbf{p}), 1) \tag{5}$$

$$\mathcal{L}_{joint} = \|\mathbf{w}_{\sigma_w}(\mathbf{x}_{joint}) - \mathbf{w}_{joint,target}\|_2^2 \tag{6}$$

where $\mathbf{w}_{joint,target}$ is a vector that is 0.5 for the neighboring bones and 0 elsewhere. These two additional losses help to bootstrap training and are applied only during the first epoch. We set the weights of the losses to $\lambda_{BCE} = 1, \lambda_{bone} = 1, \lambda_{joint} = 10$ in the first epoch and to $\lambda_{BCE} = 1, \lambda_{bone} = 0, \lambda_{joint} = 0$ afterwards.

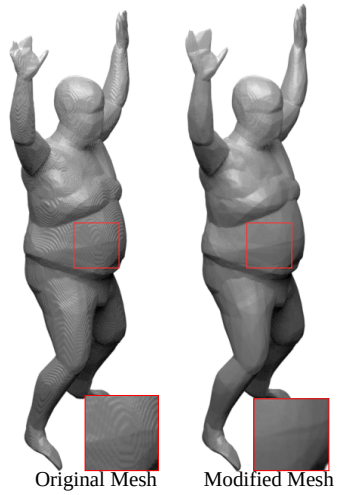
1.4. Training

We train our network using the Adam optimizer [7] with a learning rate of $\eta = 10^{-4}$ without weight decay or learning rate decay. For other hyper-parameters of Adam, the default values are used: $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Training a model takes around 36h on a single RTX 2080Ti GPU.

1.5. Mesh Extraction

Following NASA [5], we adopt *Multiresolution IsoSurface Extraction* [10] to extract meshes from the continuous occupancy fields in the deformed space. We use the same spatial resolution for NASA's and our results. The qualitative results in the supplementary video and this document are generated using a grid resolution of 512^3 and 256^3, respectively. The inference time for extracting one mesh is around 1 minute at 512^3 resolution and around 10 seconds at 256^3 resolution.

We note that in NASA's official implementation, the mesh surface is extracted as the 0.5-level set of the predicted occupancy probability, which differs from the common practice [10], where the iso-surface is extracted as the 0-level set of the predicted logits (raw output of the occupancy network without applying the sigmoid activation). This leads to noticeable step artifacts in NASA's results (shown on the right) as the marching cubes algorithm requires a smooth transition near the occupancy boundary to interpolate the positions of triangle vertices while the sigmoid activation function increases the slope near the occupancy boundary. For a fair qualitative comparison, we thus modify the mesh generation code in NASA to obtain qualitative results that do not suffer from strong discretization artifacts as shown on the right.



NASA's Result

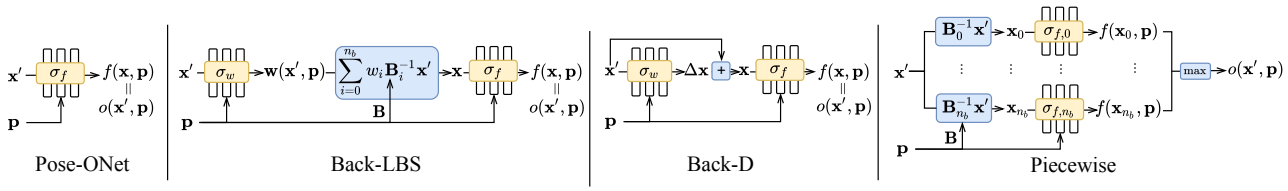


Figure 2: **Illustration of Baseline Methods.** Pose-ONet takes a deformed point $\mathbf{x}'$ and the pose as input, and directly output the corresponding occupancy probability. Back-LBS first predicts the skinning weights of a point in deformed space with a pose-conditioned skinning network. Subsequently, it determines the canonical correspondence via LBS, and finally outputs the occupancy of the canonical correspondence as the occupancy of the deformed point. Back-D directly predicts the displacement from deformed space to canonical space. The piecewise baseline transforms the query point rigidly to the canonical space of each bone, and then obtains multiple occupancy predictions, one from each bone occupancy network. These occupancy predictions are then aggregated via a max operator.

2. Evaluation Protocol

In this section, we provide additional details about our evaluation protocol.

2.1. Dataset

For within-distribution evaluation on 3D minimally clothed humans, we follow NASA and use 10 subjects from the DFaust [3] subset of AMASS [9]. For each subject, we split 10 sequences into 9 sequences for training and 1 held-out sequence for testing. In total, the training set for each subject contains 3000 training samples on average, and the test set contains 340 samples on average.

For out-of-distribution evaluation, we test the DFaust-trained models' performance on poses from the PosePrior [1] dataset. We select 7 sequences which contain full-body movements (op2, op3, op4, op5, op7, op8, op9). For each sequence, we evaluate on every 10th frame, resulting in 1096 evaluation frames in total. The ground truth meshes are generated by feeding pose parameters from PosePrior and shape parameters from respective subjects in DFaust to SMPL [8]. The ground-truth occupancy probabilities are obtained by determining whether points lie inside the ground-truth meshes. We evaluate each subject's model on all 7 test sequences and report the average score across.

2.2. Baselines

We illustrate our baseline methods in Fig. 2. We use the same set of hyper-parameters and network architectures for our self-implemented baselines as described in Sec. 1. For backward skinning (Back-LBS), we extend the skinning network to take the pose condition as input. Similar to how the pose is injected into the occupancy network, we reduce the pose condition's dimension to 8 using a linear layer (which is jointly trained with the other network parameters) and concatenate this 8-dimensional embedding to the input of the skinning network. For backward displacement (Back-D), the displacement network shares the same design as the backward skinning network, except for the output dimension. For NASA, we run the official implementation using the default hyper-parameters[1].

3. Supplementary Results

3.1. Additional results on 3D Minimally Clothed and Clothed Humans

We show additional qualitative comparisons with baselines on 3D minimally clothed humans in Fig. 3. Besides, we show more qualitative results of various different subjects in diverse poses with clothing (Fig. 5) and minimal clothing (Fig. 4) produced by our method. Finally, we demonstrate the learned skinning weights in Fig. 6.

3.2. Results on 3D Animals

As proof of concept and to demonstrate the flexibility of our approach, we also trained our model on 3D animal shapes. We obtain the training data, namely meshes in different poses and the associated bone transformations, by randomly posing the SMAL model [14], a parametric model of animals. As shown in Fig. 7, our model is able to faithfully recover skinning weights for animal shapes by learning from the posed meshes without surface correspondence or skinning weights ground truth. It further generates plausible shapes for different poses.

[1] https://github.com/tensorflow/graphics/tree/master/tensorflow_graphics/projects/nasa

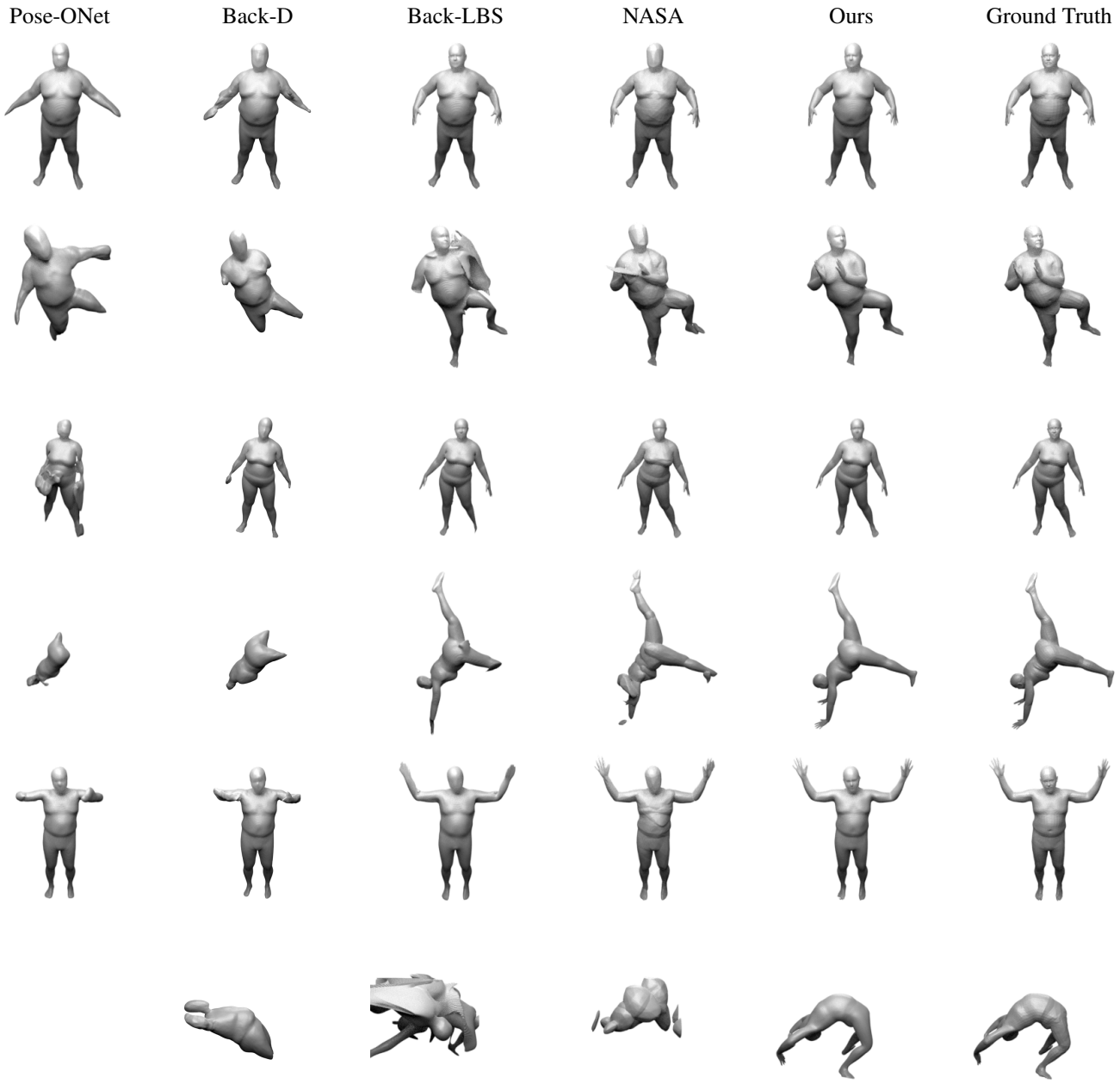


Figure 3: **Qualitative Comparison to Baselines on Minimally Clothed Humans.** Consecutive rows show two different poses of the same subject: one pose within the data distribution in the first row and one pose outside the data distribution in the second row. For some samples, Pose-ONet fails to produce meshes, hence the corresponding entries are blank.

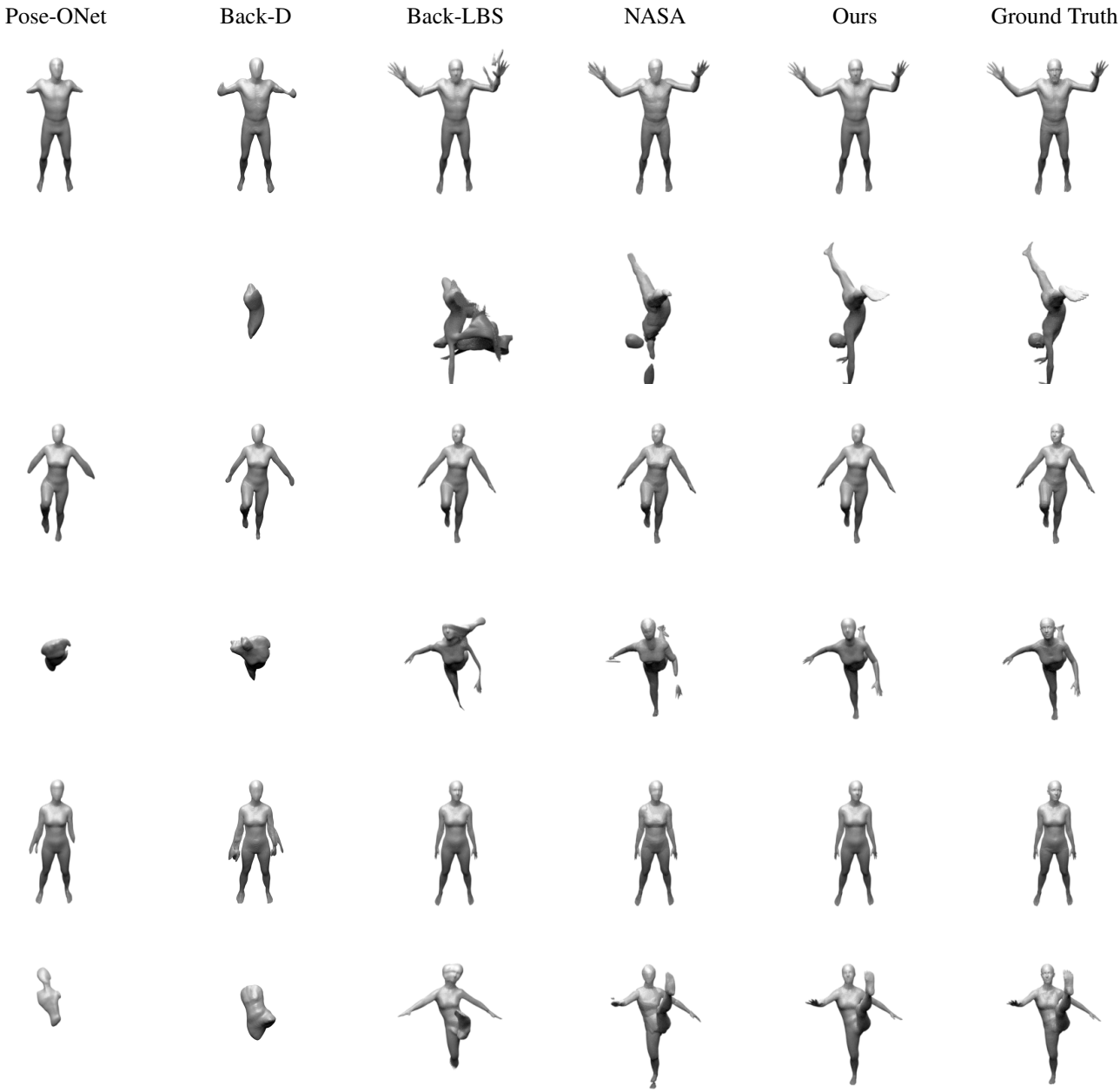


Figure 3: **Qualitative Comparison with Baselines on Minimally Clothed Humans.** Consecutive rows show two different poses of the same subject: one pose within the data distribution in the first row and one pose outside the data distribution in the second row. For some samples, Pose-ONet fails to produce meshes, hence the corresponding entries are blank.

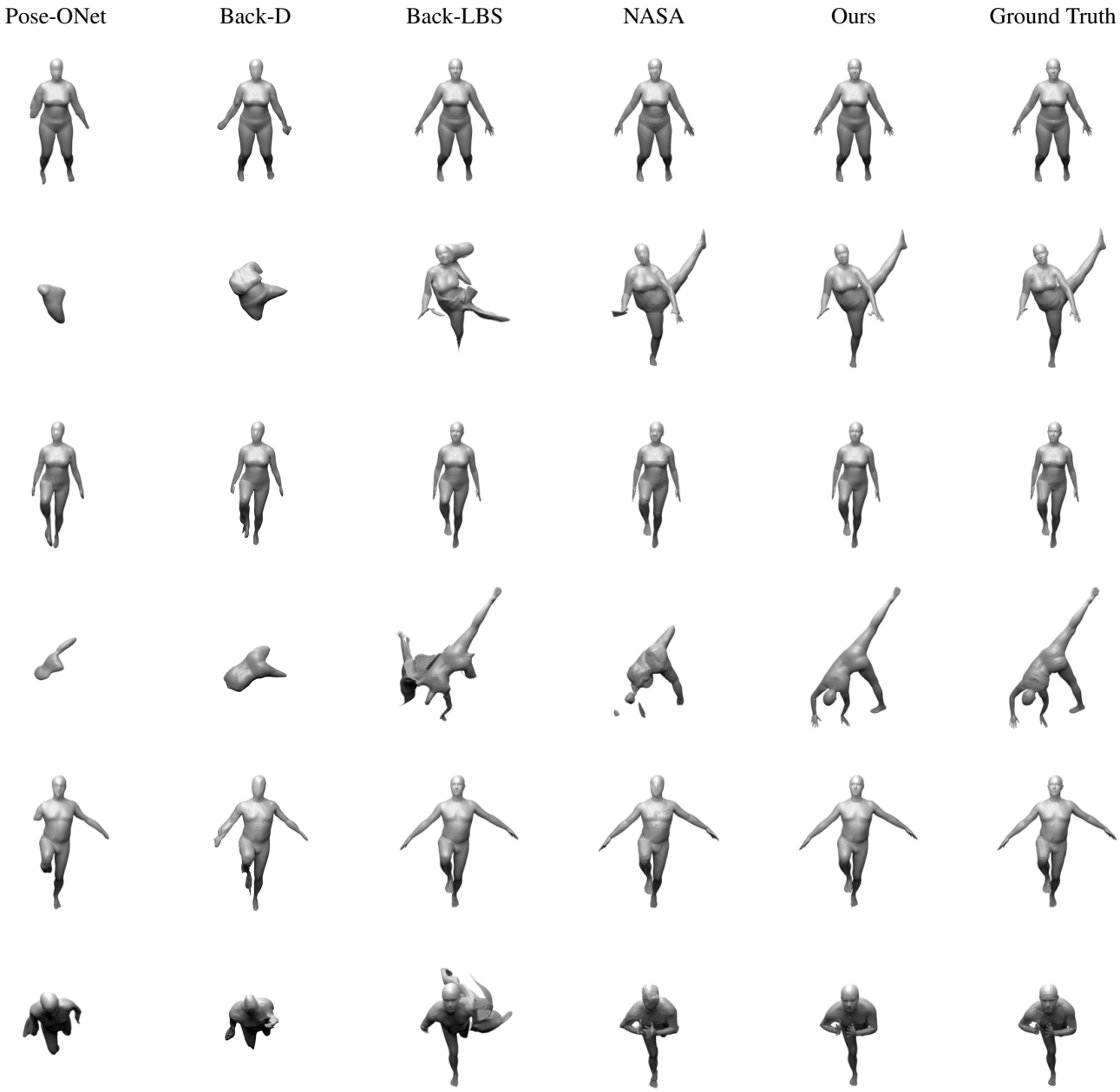


Figure 3: **Qualitative Comparison to Baselines on Minimally Clothed Humans.** Consecutive rows show two different poses of the same subject: one pose within the data distribution in the first row and one pose outside the data distribution in the second row. For some samples, Pose-ONet fails to produce meshes, hence the corresponding entries are blank.

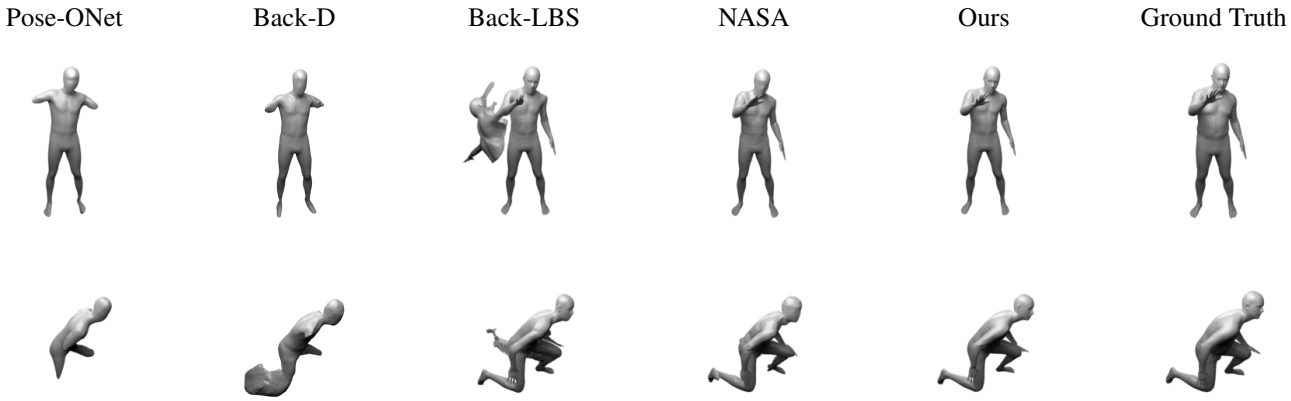


Figure 3: **Qualitative Comparison to Baselines on Minimally Clothed Humans.** Consecutive rows show two different poses of the same subject: one pose within the data distribution in the first row and one pose outside the data distribution in the second row. For some samples, Pose-ONet fails to produce meshes, hence the corresponding entries are blank.

Figure 4: **Qualitative Results on Minimally Clothed Humans.** Consecutive rows show the same subject in diverse poses outside the training distribution.

Figure 4: **Qualitative Results on Minimally Clothed Humans.** Consecutive rows show the same subject in diverse poses outside the training distribution.

Figure 4: **Qualitative Results on Minimally Clothed Humans.** Consecutive rows show the same subject in diverse poses outside the training distribution.

Figure 4: **Qualitative Results on Minimally Clothed Humans.** Consecutive rows show the same subject in diverse poses outside the training distribution.

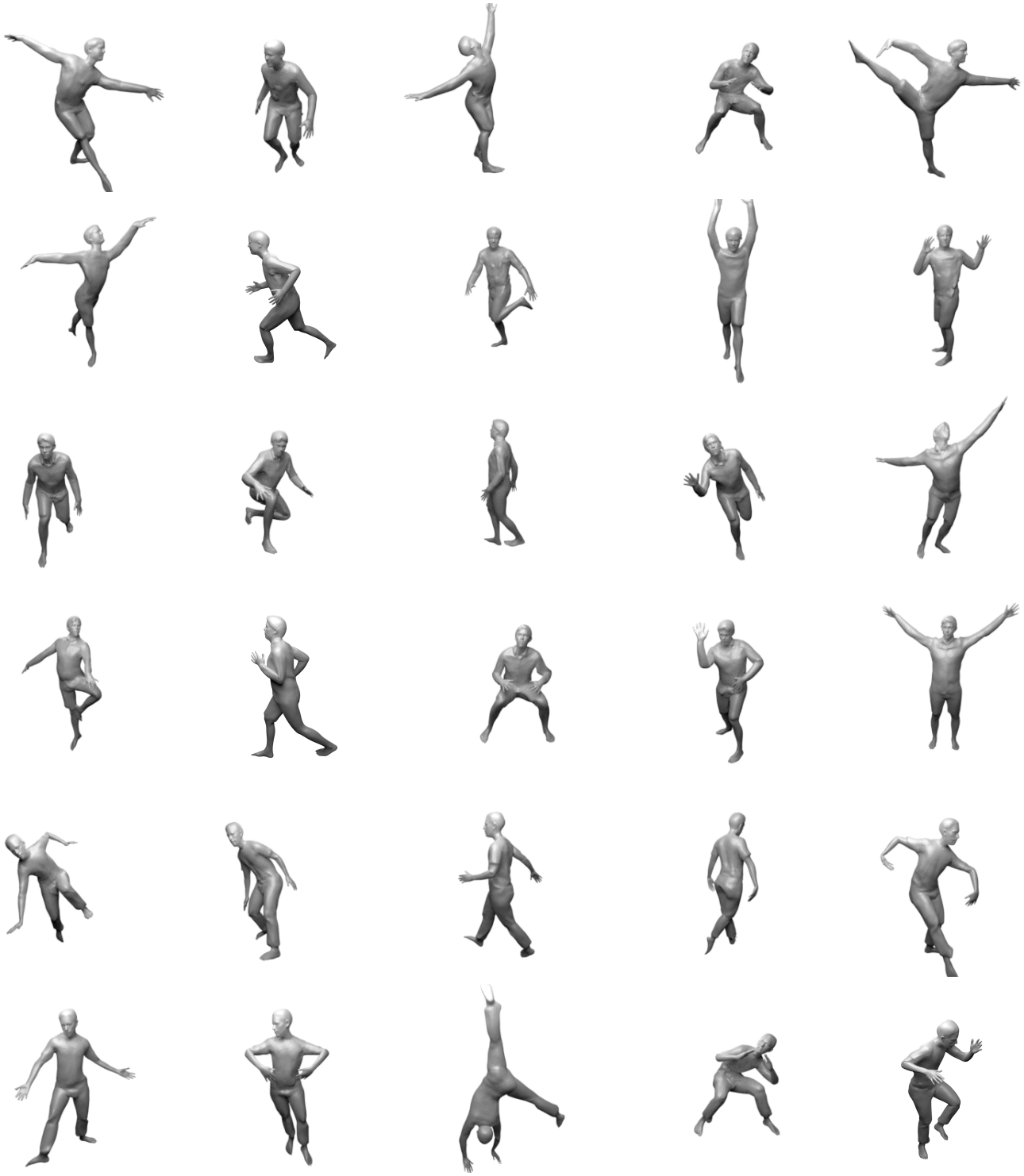


Figure 5: **Qualitative Results on Clothed Humans.** Consecutive rows show the same subject in diverse poses outside the training distribution.

Figure 5: **Qualitative Results on Clothed Humans.** Consecutive rows show the same subject in diverse poses outside the training distribution.

Figure 5: **Qualitative Results on Clothed Humans.** Consecutive rows show the same subject in diverse poses outside the training distribution.

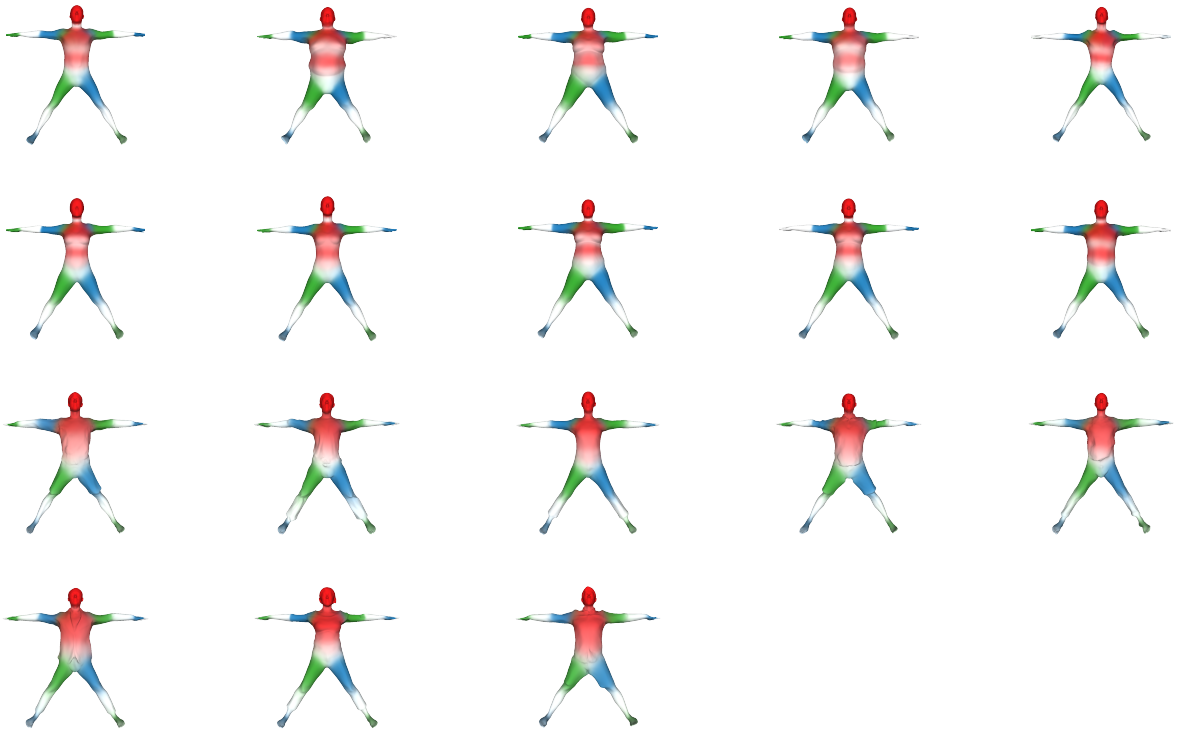


Figure 6: **Visualization of Learned Skinning Weights.** Skinning weights are overlayed on meshes extracted from the canonical occupancy network with a random pose condition. Skinning weights for minimally clothed subjects are shown in the first two rows, and skinning weights for clothed subjects are shown in the last two rows.

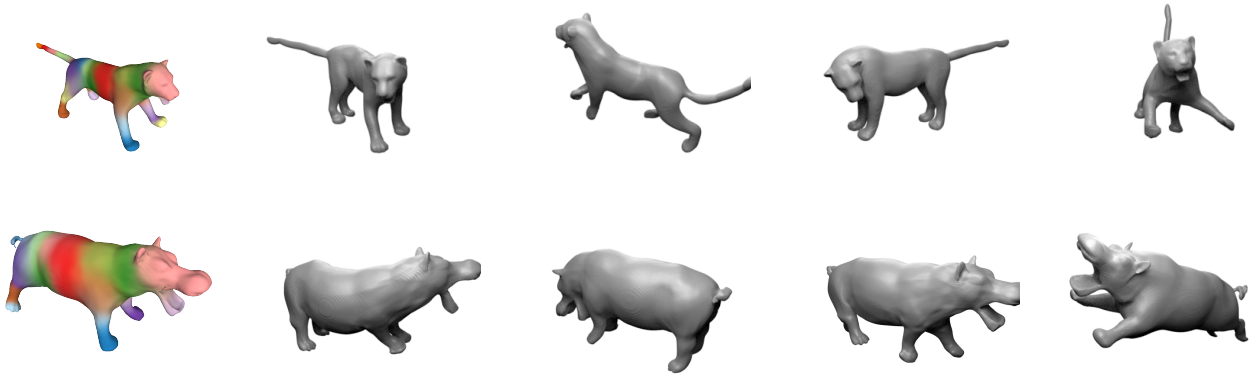


Figure 7: **Qualitative Results on Animals.** The first column shows the learned skinning weights in canonical space.

References

[1] Ijaz Akhter and Michael J. Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[2] Matan Atzmon and Yaron Lipman. SAL: Sign agnostic learning of shapes from raw data. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1

[3] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[4] Charles G Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92):577–593, 1965. 2

[5] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Neural articulated shape approximation. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 2

[6] Charles Dugas, Yoshua Bengio, François Bélisle, Claude Nadeau, and René Garcia. Incorporating second-order functional knowledge for better option pricing. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2000. 1

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2015. 2

[8] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. on Graphics*, 2015. 3

[9] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 3

[10] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[11] 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 *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 1

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1

[13] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 1

[14] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3