Neural Point-based Shape Modeling of Humans in Challenging Clothing **Supplementary Material**

Qianli Ma[1,2] Jinlong Yang[2] Michael J. Black[2] Siyu Tang[1]
[1]ETH Zürich [2]Max Planck Institute for Intelligent Systems, Tübingen, Germany
{qianli.ma, siyu.tang}@inf.ethz.ch, {qma,jyang,black}@tuebingen.mpg.de

In this supplementary document, we provide more details on the model designs and experimental setups (Sec. S1), further elaborations on our method (Sec. S2), and further discussions on experimental results, limitations and failure cases (Sec. S3). Please also refer to the supplementary video for animated results.

S1. Implementation Details

S1.1. SkiRT

Architecture. The coarse and the fine shape prediction have the same architectural design, but majorly differ in their inputs. Both networks are a 8-layer multi-layer perceptron (MLP) comprising 256 neurons per layer, and with the SoftPlus as the nonlinear activation. Following [7], a skip connection is added from the input layer to the fourth layer of the network. From the fifth layer the network branches out two heads with the same architecture that predicts the point locations and normals respectively.

The coarse shape MLP takes as input the Cartesian coordinates of a query point from the surface of the SMPL-X body in the canonical pose $\hat{b}_i \in \mathbb{R}^3$, as well as a global shape code $z^{\text{coarse}} \in \mathbb{R}^{256}$. The global shape code is shared by all query locations for the coarse stage.

The fine detail MLP takes a query coordinates $\hat{b}_i \in \mathbb{R}^3$, a local shape descriptor $z_i^{\text{fine}} \in \mathbb{R}^{64}$, and a local pose descriptor $z_i^{\text{pose}} \in \mathbb{R}^{64}$. Each query location on the body surface is associated with a unique local shape descriptor z_i^{fine} that is an optimizable latent vector learned in an auto-decoding fashion [7]. To get the pose descriptor z_i^{pose}, we follow [6] and first encode the UV-positional map (with a resolution of 128×128) of the unclothed posed body by a U-Net [9], resulting in a $128 \times 128 \times 64$-dimensional pose feature tensor. For each query location $\hat{b}_i$, we find its corresponding UV-coordinate $\hat{u}_i \in \mathbb{R}^2$, and obtain the local pose feature z_i^{pose} by querying the spatial dimensions (i.e. the first two dimensions) of the pose feature tensor using $\hat{u}_i$ with bilinear interpolation. We empirically find that this U-Net-based pose feature encoder provides the sharpest geometric details. Replacing it with e.g. the filtered local pose parameters (as in SCANimate [10]) yields a significant performance drop.

The skinning weights prediction network is a 5-layer MLP that comprises 256 neurons per intermediate layer. The final layer outputs a 22-dimentional vector that corresponds to the 22 clothing-related body joints in the SMPL-X model (we merged all finger-related joints into a "hand" joint for each hand). LeakyReLU is used as the nonlinear activation except for the last layer where a SoftMax is used to obtain normalized skinning weights.

Loss functions. In addition to the loss functions introduced in the main paper Eqs. (8)-(10), we follow [6] and use Chamfer Distance and the normal loss to train our model.

The bi-directional Chamfer Distance penalizes the L2 discrepancy between the generated point cloud $\mathbf{X} = \{x_i\}$ and the ground truth $\mathbf{Y} = \{y_j\}$: $\mathcal{L}_{\text{CD}} =$

$$\frac{1}{|\mathbf{X}|}\sum_{i=1}^{|\mathbf{X}|}\min_j \|x_i - y_j\|_2^2 + \frac{1}{|\mathbf{Y}|}\sum_{j=1}^{|\mathbf{Y}|}\min_i \|x_i - y_j\|_2^2. \quad (1)$$

The normal loss $\mathcal{L}_{\text{n}}$ is the L1 discrepancy between each point's predicted unit normal $\boldsymbol{n}(x_i)$ and the normal of its nearest neighbor in the ground truth point cloud:

$$\mathcal{L}_{\text{n}} = \frac{1}{|\mathbf{X}|}\sum_{i=1}^{|\mathbf{X}|}\left\|\boldsymbol{n}(x_i) - \boldsymbol{n}(\underset{y_j \in \mathbf{Y}}{\operatorname{argmin}}\, d(x_i, y_j))\right\|_1. \quad (2)$$

Note that the normal loss is computed based on the nearest points found by the Chamfer Distance, intuitively it is more effective when the point locations of the predicted point cloud roughly matches the ground truth. Therefore we introduce the normal loss in our training after 100 epochs when the Chamfer loss plateaus.

The total loss is a weighted sum of all these terms:

$$\mathcal{L} = \lambda_{\text{CD}}\mathcal{L}_{\text{CD}} + \lambda_{\text{n}}\mathcal{L}_{\text{n}} + \mathcal{L}_{\text{rgl}} + \lambda_{\text{LBS}}\mathcal{L}_{\text{LBS}} + \lambda_{\text{reproj}}\mathcal{L}_{\text{reproj}}. \quad (3)$$

The weights are set as $\lambda_{\text{CD}} = 1e4, \lambda_n = 1.0, \lambda_{\text{LBS}} = 1.0$, $\lambda_{\text{reproj}} = 5e2$. Note that, as discussed in the main paper, the displacement regularization term $\mathcal{L}_{\text{rgl}}$ contains a

per-point adaptive weight λ_i^{adapt} which has an initial value of $2e3$ but decays at each point adaptively according to its kNN radius. See Sec. S2.2 for more details.

Training and Inference. We use query points from the low-resolution (128×128) UV map to train the coarse stage MLP, and evaluate it at test-time with denser query locations (e.g. those from a 256×256 UV map) that match the number of points on the fine stage. Although it is also possible to train the coarse stage with denser points too, we find that the resulting coarse shape tends to be noisy. Using fewer points for training prevents the model from being overfit to local details of certain training examples, and yields a smoother learned coarse shape. The coarse stage is trained using the Adam optimizer with a learning rate of $3e-4$ for 150 epochs, which takes approximately 0.9 hour on a NVIDIA RTX 6000 GPU.

For the fine stage, we train the displacement network and the LBS MLP together using Adam with a learning rate of $3e-4$ for 250 epochs, which takes approximately 2.5 hours on a NVIDIA RTX 6000 GPU.

S1.2. Baselines

SCANimate. For the pose-dependent shape prediction task (main paper Sec. 5.1), we use the official implementation and hyperparameter settings from SCANimate [10] and train a separate model for each outfit in the ReSynth dataset [6]. Since SCANimate requires watertight meshes for training the implicit surfaces, we process the dense point clouds provided in the ReSynth dataset into watertight meshes using Poisson Surface Reconstruction [3] and then use this processed data as ground truth to train the model. During training, we sample 8000 points from the clothed body meshes dynamically at each iteration.

At test-time, for a fair comparison with the point-based methods, we first extract a surface from the SCANimate's implicit predictions using Marching Cubes [4], and then sample the same number (47,911) of points from the surface for evaluating the quantitative errors.

POP. We train and evaluate POP also in a subject-specific manner to fairly compare with all other methods. Following the official implementation, we train and evaluate model using a fixed set of 47911 query points on the body that correspond to the valid pixels from the 256×256 UV-map of SMPL-X. The reported quantitative errors are averaged over all the query points.

S1.3. Further Details on Experimental Setups

Here we provide more details on scan completion experiment as described in the main paper Sec. 5.3. We scan subjects wearing challenging clothing (e.g. skirts and very loose and wrinkly shirts) using a body scanner. The subjects are asked to improvise on several motion sequences, resulting in few hundreds of frames (typically 200-300) of raw scan data. We then fit the SMPL-X body model to the scan data, and train a subject-specific SkiRT model with the scan-body pairs using the same procedure and hyperparameters as described in Sec. S1. The subjects' hands and feet are often largely incomplete throughout the sequence in the captured raw scans due to the hardware limitations. Therefore we zero out the displacement predictions for these parts. At test time, we feed the model with the training bodies. The trained model is expected to reproduce the training data and outputs the hole-filled complete scans of point clouds.

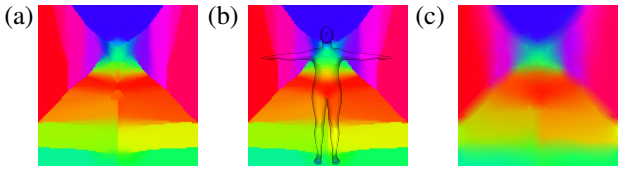

Figure S1: Visualizing a slice from the pre-diffused SMPL-X LBS weight field. (a) Diffusion using nearest neighbor assignment. (b) Overlay of the SMPL-X body (color-coded with the body LBS weights) in the nearest-neighbor-diffused field. (c) The smoothed field after optimization convergence. The colors in the fields indicate the skinning weights to body part with the corresponding color in (b).

S2. Extended Illustrations

S2.1. Smoothing the LBS Weight Field

As described in the main paper Sec. 4.1, we first follow LoopReg [2] and diffuse LBS weights of the SMPL-X model to $\mathbb{R}^3$ the using nearest neighbor assignment: for each query point in $\mathbb{R}^3$, we find its nearest point on the SMPL-X body surface and assign its LBS weight to the query point. In practice, the diffused LBS weights are precomputed on a regular grid (can also be seen as voxel centers) with a resolution of $128 \times 128 \times 128$. For an arbitrary query point, its LBS weights are acquired via trilinear interpolation on its neighboring grid points. As shown in Fig S1(a), the nearest neighbor assignment results in clear discontinuities of the LBS weight distribution in space. A similar illustration can also be found in Fig. 3 in [2].

To smooth the spatial distributions of the LBS weights, we use an optimization-based approach. On a high level, for each grid point, we minimize the discrepancy between its LBS weights and the average of its 1-ring neighbors. Formally, we optimize the LBS weights on all the grid points with respect to the following energy function:

$$\mathcal{E}_{\text{smooth}} = \sum_{i=1}^{M} \Big\| w_{p_i} - \frac{1}{|\mathcal{N}(p_i)|} \sum_{\forall q_j \in \mathcal{N}(p_i)} w_{q_j} \Big\|_1, \quad (4)$$

where p_i denotes a grid point, $\mathcal{N}(p_i)$ denotes the set of its 1-ring neighbors, q_j is a point in the neighborhood, and M is the total number of grid points. The optimization effectively smoothes the boundary discontinuities of the LBS weights in space, as shown in Fig. S1(c).

S2.2. Point-adaptive Regularization/Upsampling

As described in the main paper Sec. 4.3, both regularization (for training) and the point-adaptive upsampling (for post-processing) are based on the kNN radius l_i , i.e. the averaged k-nearest neighbor (we used k=5) distance, per point. Let μ_l and σ_l be the mean and standard deviation of points' kNN radius calculated over the entire point set. During training, if a point's kNN radius is above a threshold of mean plus two standard deviations, we disable the regularization on the normal of the displacement for that point:

$$\lambda_i^{\text{adapt}} = \begin{cases} 0 & \text{if } l_i > \mu_l + 2\sigma_l, \\ 2e3 & \text{otherwise.} \end{cases} \tag{5}$$

We also experimented with more sophisticated policy such as decaying the per-point weight using an exponential decay with respect to the kNN radius, but do not observe noticeable improvements.

The simple thresholding in Eq. (6) is also applied to the upsampling in the post-process. Consider a point x_i with $l_i > \mu_l + 2\sigma_l$, we aim to generate more points around it. To do so, we find the triangle on the SMPL-X body mesh where the x_i's corresponding body point $\hat{b}_i$ locates, and assign a higher sampling weight η_i for the triangle:

$$\eta_i^{\text{resample}} = \begin{cases} 2^{\frac{l_i - \mu}{\sigma}} & \text{if } l_i > \mu_l + 2\sigma_l, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

We then re-sample query points from the body mesh surface by modifying the standard uniform mesh sampling. In the uniform sampling, the sampling probability on each triangle is proportional to its area. We scale this probability with η_i^{resample}, so that more points are generated on the regions that originally have lower point density. The resampled query points are then fed into the MLP to generate the new points on the clothing surface. The final result is the union of the new points and the original point set. This process can be repeated for several iterations for improved visual quality. The results shown in the main paper (third column in Fig. 5) uses 3 iterations of upsampling.

S3. Extended Results and Discussions

S3.1. Limitation and Failure Cases

In the case of very loose dresses, the points on the dress surface produced by SkiRT can still be visibly sparser than on other body parts as visualized in the main paper Fig. 5; consequently the mesh reconstruction can have rough surfaces as shown in Fig. S2.

Although the coarse stage enables handling challenging clothing types, the point-based coarse geometry is typically not as smooth as that of the base body mesh. Consequently, when adding the fine-stage predictions to the coarse shape, the final geometry can possibly be noisier than POP (which adds displacements directly to the unclothed body), hence occasional loss of clothing details (as seen in the main paper Fig. 5) and a noisier surface reconstruction (as shown in Fig. S2), despite a higher quantitative accuracy on entire test set. Unlike the mesh representation where one could apply loss functions (such as the Laplacian term) that encourage the smoothness in the generated geometry, for point clouds such regularization is not trivial due to the absence of the point connectivity, and we leave this for future work.

As with other recent models [6, 10], our model requires the fitted underlying body to be accurate. While this is not an issue with the synthetic ReSynth data that we use in the paper, we observe certain failure cases with the real data where the estimated body under clothing is imperfect. For example, in the scan completion experiment, the fitted body can be inaccurate due to the challenging clothing. Consequently, in the resulting completed scans flickering can be observed especially at the extremities, as shown in the supplementary video. In the future we plan to refine the body pose together with the network parameters during training so as to better handle real world data.

S3.2. Mesh Reconstruction

In Fig. S2 we qualitatively show the results of applying Poisson Surface Reconstruction (PSR) to the point sets generated by a baseline method and ours, respectively. Through building an implicit field of the indicator function, the PSR is capable of filling holes in the point clouds to a certain extent. However, when the point cloud has obvious discontinuities as with the previous method [6], the reconstructed meshes are prone to artifacts. In contrast, our method generates point clouds that have a more uniform distribution of points on the clothing surface, and thus effectively alleviates this issue.

Note that the purpose of this experiment is to show the impact of the split-like artifacts on the quality of mesh reconstruction. However, mesh reconstruction is not the sole exit of point-based human representations. Recent work has shown the potential of combining the point-based representation with neural rendering to achieve realistic renderings without the intermediate meshing step [1, 5, 8]. Combining these methods with our pipeline is an promising direction for future research.

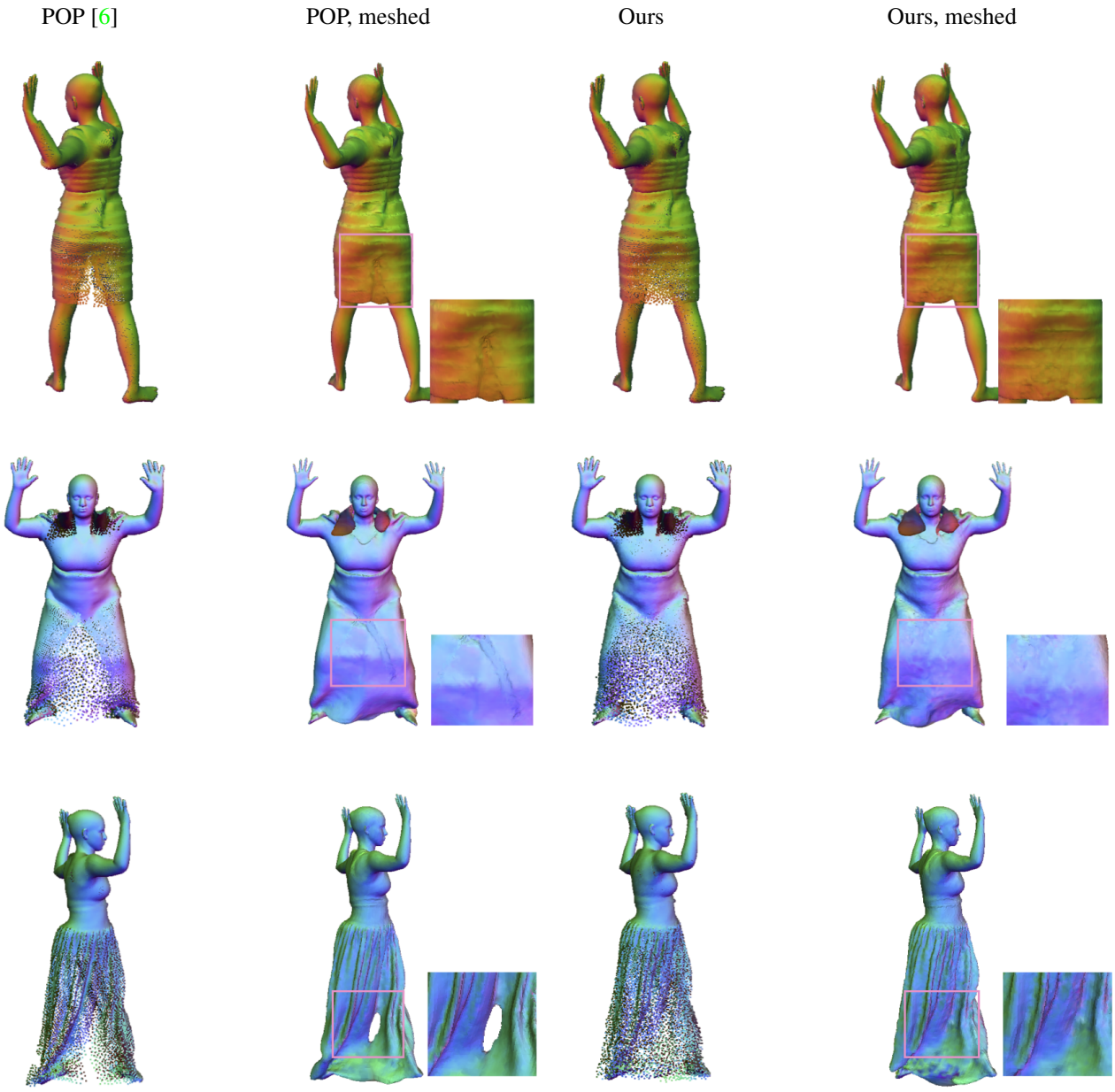

Figure S2: Comparison between the point-based baseline POP [6] and our method in terms of mesh reconstruction quality. The meshed results are produced by performing Screened Poisson Surface Reconstruction [3] on the corresponding point clouds. When the clothed body point cloud has clear gaps on the clothing surface, the reconstructed mesh is prone to artifacts.

S3.3. Scan Completion

Please refer to the supplementary video for the animated results of the scan completion experiment.

References

[1] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 696–712, 2020. 3

[2] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. LoopReg: Self-supervised learning of implicit surface correspondences, pose and shape for 3D human mesh registration. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12909–12922, 2020. 2

[3] Michael Kazhdan and Hugues Hoppe. Screened Poisson sur-

face reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):1–13, 2013. 2, 4

[4] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *ACM SIGGRAPH Computer Graphics*, volume 21, pages 163–169, 1987. 2

[5] Qianli Ma, Shunsuke Saito, Jinlong Yang, Siyu Tang, and Michael J. Black. SCALE: Modeling clothed humans with a surface codec of articulated local elements. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 16082–16093, June 2021. 3

[6] Qianli Ma, Jinlong Yang, Siyu Tang, and Michael J. Black. The power of points for modeling humans in clothing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10974–10984, Oct. 2021. 1, 2, 3, 4

[7] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 165–174, 2019. 1

[8] Sergey Prokudin, Michael J. Black, and Javier Romero. SMPLpix: Neural avatars from 3D human models. In *Winter Conference on Applications of Computer Vision (WACV)*, 2021. 3

[9] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015. 1

[10] Shunsuke Saito, Jinlong Yang, Qianli Ma, and Michael J. Black. SCANimate: Weakly supervised learning of skinned clothed avatar networks. In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2021. 1, 2, 3