SCALE: Modeling Clothed Humans with a Surface Codec of Articulated Local Elements
Supplementary Material

Qianli Ma[1,2] Shunsuke Saito[1] Jinlong Yang[1] Siyu Tang[2] Michael J. Black[1]
[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany [2]ETH Zürich

S1. Implementation Details

S1.1. Network Architectures

To encode our UV positional map of resolution 32×32 into local features, we use a standard UNet [11] as illustrated in Fig. S1(a). It consists of five [Conv2d, BatchNorm, LeakyReLU(0.2)] blocks (red arrows), followed by five [ReLU, ConvTranspose2d, BatchNorm] blocks (blue arrows). The final layer does not apply BatchNorm.

To deform the local elements, we use an 8-layer MLP with a skip connection from the input to the 4th layer as in DeepSDF [9], see Fig. S1(b). From the 6th layer, the network branches out three heads with the same architecture that predicts residuals from the basis point locations, normals and colors respectively. Batch normalization and the SoftPlus nonlinearity with $\beta = 1$ are applied for all but the last layer in the decoder. The color prediction branch finishes with a Sigmoid activation to squeeze the predicted RGB values between 0 and 1. The predicted normals are normalized to unit length.

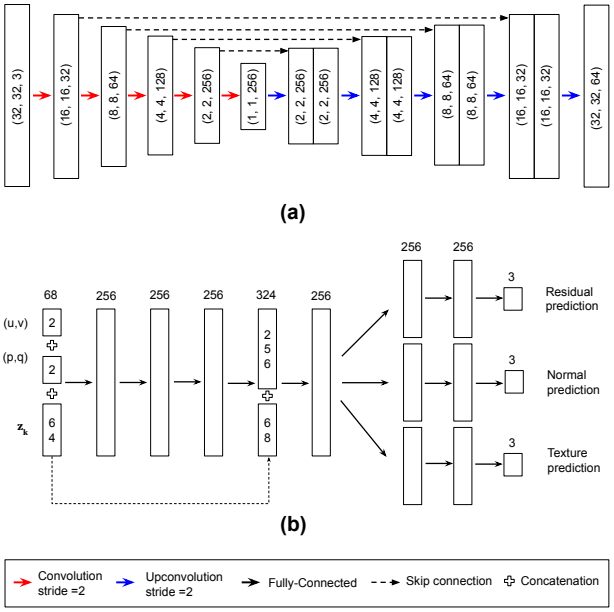

Figure S1: A visualization of our network architectures. **(a)** The UNet for our UV pose feature encoder. **(b)** The MLP for patch deformations. The numbers denote the dimensions of the network input or the layer outputs.

S1.2. Training and Inference

We train SCALE with the Adam [6] optimizer with a learning rate of $3.0e-4$, a batch size of 16, for 800 epochs. As the early stage of the training does not reliably provide nearest neighbor points on the ground-truth, we add $\mathcal{L}_{\text{n}}$ and $\mathcal{L}_{\text{c}}$ when $\mathcal{L}_{\text{d}}$ roughly plateaus after 250 epochs.

The residual, normal and color prediction modules are trained jointly. To balance the loss terms, the weights are set to $\lambda_{\text{d}} = 2e4, \lambda_{\text{r}} = 2e3, \lambda_{\text{c}} = \lambda_{\text{n}} = 0$ at the beginning of the training, and $\lambda_{\text{c}} = \lambda_{\text{n}} = 0.1$ from the 200th epoch when the point locations are roughly converged.

For the inference time comparison in the main paper Tab. 1, we report the wall-clock time using a desktop workstation with a Xeon CPU and Nvidia P5000 GPU.

S1.3. Data Processing

We normalize the bodies by removing the body translation and global orientation from the data. The motion sequences are randomly split into train (70%) and test (30%) sets. For the clothing types in the main paper, the number of train / test data samples is: *blazerlong* 1334 / 563; *shortlong* 3480 / 976; and *skirt* 5113 / 2022.

S1.4. Definition of the Local Coordinates

Here we elaborate on the local coordinate system used in the main paper Eq. (5). As illustrated in Fig. S2, for each body point $\mathbf{t}_k$, we find the triangle where $\mathbf{t}_k$ sits on the SMPL [7] body mesh. We take the first two edges $\vec{e}_{k1}, \vec{e}_{k2}$ of the triangle, as well as the normal vector of the triangle plane $\vec{e}_{k3} = \vec{e}_{k1} \times \vec{e}_{k2}$, as three axes of the local coordinate frame. Note that $\vec{e}_{k1}, \vec{e}_{k2}, \vec{e}_{k3}$ are unit-length column vectors. The transformation associated with $\mathbf{t}_k$ is then defined

as: $\mathbf{T}_k = [\vec{e}_{k1}, \vec{e}_{k2}, \vec{e}_{k3}]$. The residual predictions $\mathbf{r}_k$ from the network are relative to the local coordinate system, and are transformed by $\mathbf{T}_k$ to the world coordinate according to the main paper Eq. (5).

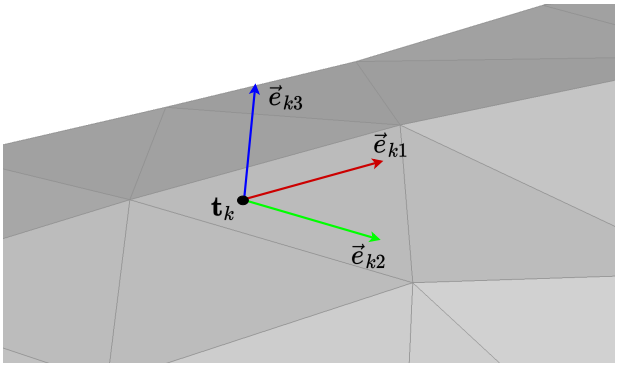

Figure S2: An illustration of the local coordinate system defined on a body point $\mathbf{t}_k$. We take the triangle where it locates on the SMPL body mesh (in grey), and build the local coordinate frame using the edges and surface normal of the triangle.

S1.5. Adaptive Upsampling

During inference, SCALE allows us to sample arbitrarily dense points to obtain high-resolution point sets. As the UV positional map provided by the SMPL model [7] has a higher density around the head region and lower density around the legs, we mitigate the problem of unbalanced point density by resampling points proportional to the area of each local element. Note that we approximate the area of patches by summing the areas of triangulated local grid points. See Sec. S2.2 for qualitative results of the adaptive sampling.

S1.6. Neural Rendering

Elaborating on the neural rendering of SCALE as shown in the main paper Sec. 4.5, we use the SMPLpix [10] model for neural rendering. It takes as input an RGB-D projection of the colored point set generated by SCALE, and outputs a hole-filled, realistic, image of the predicted clothed human.

RGB-D projections. Given the colored point set, $\mathbf{X}^+ = [\mathbf{X}, \mathbf{X}^c] \in \mathbb{R}^{KM\times 6}$, where $\mathbf{X}^c \in \mathbb{R}^{KM\times 3}$ are the RGB values of the points $\mathbf{X}$, we perform 2D projections using a pre-defined set of camera parameters $(\mathbf{K}, \mathbf{R}, \mathbf{t})$. The result is a set of RGB-D images, $I_{\mathbf{x}} \in \mathbb{R}^{W\times H\times 4}$. In the case where two points are projected to the same pixel, we take the value of the point that has smaller depth. These images are the inputs to the SMPLpix model.

Data and Training. We train SMPLpix using the same data and train / test split as what we use to train SCALE. Each (input, output) image pair for SMPLpix is acquired by performing the above-mentioned RGB-D projections to the SCALE predicted point set and the ground truth point set (of a higher density), respectively.

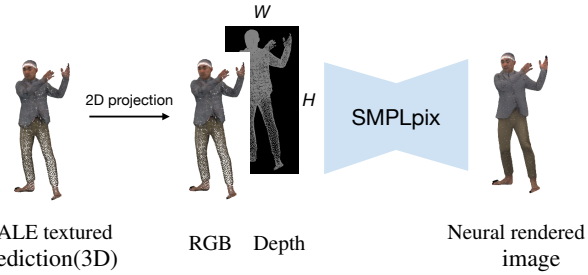

Figure S3: Pipeline of neural rendering with SMPLpix [10].

Note that the distorted fingers or toes in some of our results stem from the artifacts present in the ground truth point clouds. Similarly, the holes in the ground truth scan data lead to occasional black color predictions on these regions. In addition, as the synthetic skirt data does not have ground truth texture, we use the point normals as the RGB values for the visualization and neural rendering.

The SMPLpix network is trained with the Adam optimizer [6] with a learning rate of $1e-4$, batch size 10, for 200 epochs, using the perceptual VGG-loss [4].

Discussion. The neural point-based rendering circumvents the meshing step in traditional graphics pipelines. Our SMPLpix implementation takes on average $42ms$ to generate a 512×512 image without any hardware-specific optimization. Recall that SCALE takes less than $9ms$ to generate a set of 13K points, our SCALE+SMPLpix pipeline remains highly efficient, and shows promise for future work on image-based clothed human synthesis with intuitive pose control. Animations of the neural rendered SCALE results are provided in the supplemental video[1].

S2. Additional Discussions

S2.1. Tradeoff between Patches and Subsamples

Experiments in the main paper use $K = 798$ surface elements (which corresponds to a 32×32 UV positional map) and $M = 16$ points per element. In practice, these two numbers can be chosen per the specifications of the task.

Here we discuss a degenerated case of our general formulation: $K = 798 \times 16,\ M = 1$. That is, we use a much higher number (12,768) of surface elements (corresponding to a 128×128 UV map), and sample only one example per element, whereby the number of output points remains the same. Such a setting is equivalent to the traditional mesh vertex offset representation, where each body vertex corresponds to a point on the clothing surface.

[1] Available at https://qianlim.github.io/SCALE.

We experiment with this setting (denoted as "Vert-Offset") and compare it to our method using surface elements in Tab. S1. The number of the network parameters is kept the same for fair comparison.

The results reveal the advantage of our surface elements formulation: high fidelity and efficiency. Compared to the Vert-Offset representation, our method has $1/4$ FLOPS and lower GPU consumption in the UNet due to the $1/4$-sized UV map input. Nevertheless, it achieves overall comparable normal error and consistently lower Chamfer error.

Table S1: Comparison between our surface element formulation and vertex offset formulation. Chamfer-L_2 $\times$ $10^{-4}m^2$, normal diff $\times 10^{-1}$.

		Vert-Offset	Ours
Chamfer-L_2	*blazerlong*	1.13	**1.07**
	shortlong	0.91	**0.89**
	skirt	2.78	**2.69**
Normal diff	*blazerlong*	**1.20**	1.22
	shortlong	**1.09**	1.12
	skirt	0.97	**0.94**

S2.2. Effect of Adaptive Upsampling

Fig. S4 shows the effect of adaptive patch sampling at test time. Due to the unbalanced point density in the SMPL UV map, the SCALE output will have sparser points on the leg region if the same number of points are sampled for every surface element. Such sparse points can be insufficient to represent the complex garment geometry in these regions, e.g. in the case of skirts. Consequently, when applying Poisson Surface Reconstruction [5] to them, the reconstructed mesh will have missing geometry and ghosting artifacts, as demonstrated in the second column of Fig. S4. Adaptive patch upsampling adds more points to the bigger patches. With more points sampled on the skirt surface, the mesh reconstruction quality is improved.

The figure also shows a limitation of our model: once the model is trained, the test-time adaptive upsampling can only increase the point density *within* each patch, while the gap between patches cannot be shrunk. As discussed in the main paper, a potential solution is to more explicitly model the connectivity between the patches by incorporating a learnable triangulation. We leave this as future work.

S2.3. Extended Result Analysis

Error analysis on the patch periphery. From our qualitative results (main paper Figs. 3-5), the patches can sufficiently deform to represent fine structures such as fingers. Here, we perform additional numerical analysis by calculating the mean single-directional Chamfer error (from the predicted points to the ground truth points) of the patches' peripheral points and inner points respectively. We observe a slightly larger Chamfer error (4%) from the peripheral points than the patch center points. The low relative difference between the two is in line with the qualitative results, yet shows space for improvement in future work.

w/o adaptive points | w/o adaptive meshed | w/ adaptive points | w/ adaptive meshed | GT

Figure S4: Qualitative effects of the adaptive patch subsampling.

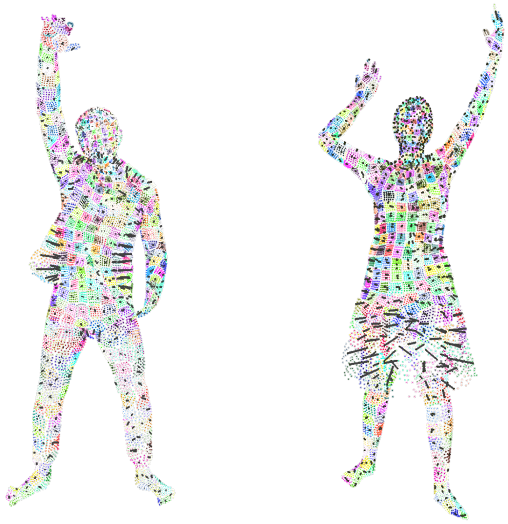

Figure S5: An illustration of the correspondence between the body points and the patches. Each black line connects a body basis point and the center of its corresponding patch.

Clothing-body correspondence. Fig. S5 illustrates the correspondence between the patches on the clothed body surface and the basis points from the underlying body.

S2.4. Additional Evaluations

CAPE data. In the main paper, we highlight the characteristics of our model on the two prototypical outfit types

(*blazerlong* and *shortlong*) from the CAPE dataset. Here we show the results on the rest of the CAPE dataset, which comprises mostly tight-fitting clothes such as short / long T-shirts, dress shirts, and short / long trousers. For each outfit, 30% of the sequences are selected for testing and the rest are for training.

As shown in Tab. S2, our model again outperforms both baselines in the Chamfer error by a large margin, and is comparable with the CAPE model in terms of normal accuracy. Note that on several sequences NASA predicts bodies with missing limbs, hence the high Chamfer error. Qualitative results in Fig. S6 are also in accordance with the main paper experiments. Results from the CAPE model [8] in general lack realistic pose-dependent clothing deformation. NASA [1] can predict detailed clothing structure with notable influences of the body pose, but often suffer from discontinuities between different body parts. SCALE produces clothing shapes that naturally move with varied poses, showing a coherent global shape and detailed local structures such as wrinkles and edges.

Table S2: Reconstruction error on the entire CAPE dataset. Chamfer-$L_2 \times 10^{-4} m^2$, normal diff $\times 10^{-1}$.

	CAPE [8]	NASA [1]	Ours
Chamfer-L_2	1.28	4.08	**0.93**
Normal diff	**1.16**	1.24	1.18

Long dress. In addition to the mid-length skirt in the main paper, we also experiment with a more challenging long dress. Similar to the skirt, the long dress data are created with physics-based simulation. Since the dress deviates from the body topology and contains thin cloth structures, both baselines, CAPE and NASA, are unable to process it. Here we compare with methods that use a global feature code with the same setting as in the main paper Sec. 4.3.

As shown in Tab. S3 and Fig. S7, for either using a large global surface element (as in AtlasNet [3] and 3D-CODED [2], the first two columns in Tab. S3) or numerous local surface elements (as in PCN [12], the last two columns in Tab. S3), decoding the shape from a global shape code in general fails to reconstruct the clothing geometry faithfully, resulting in high numerical errors. In contrast, SCALE is able to represent the wrinkles and folds on the dress while producing a smooth shape for the upper body, validating our key design choices, i.e. local feature codes and explicit articulation modeling.

S2.5. Animated Results

Please refer to the supplemental video at https://qianlim.github.io/SCALE for more qualitative comparisons against CAPE and NASA, as well as animated results produced by SCALE.

References

[1] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Neural articulated shape approximation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 612–628, 2020. 4, 5

[2] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3D-CODED: 3D correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. 4

[3] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. A papier-mâché approach to learning 3D surface generation. *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 216–224, 2018. 4, 5

[4] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 9906, pages 694–711, 2016. 2

[5] Michael Kazhdan and Hugues Hoppe. Screened Poisson surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(3):1–13, 2013. 3

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 2

[7] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics (TOG)*, 34(6):248, 2015. 1, 2

[8] 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 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 6468–6477, 2020. 4, 5

[9] 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

[10] 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. 2

[11] 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

[12] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. PCN: Point completion network. In *International Conference on 3D Vision (3DV)*, pages 728–737, 2018. 4, 5

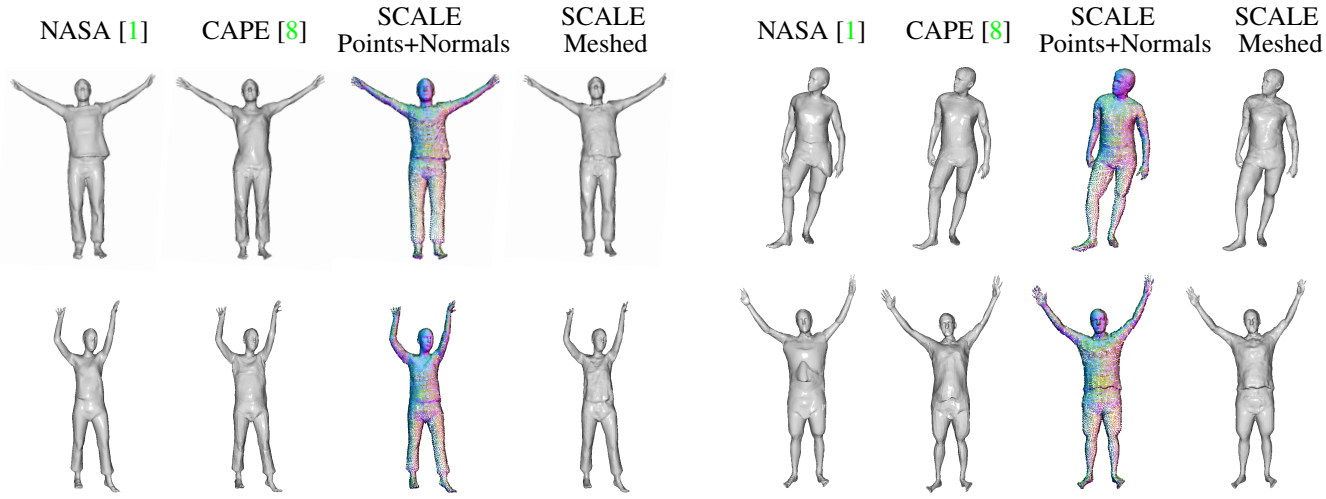

Figure S6: Extended qualitative results on the CAPE dataset.

Table S3: Comparison between our method with methods that use a global feature code on the long dress data. "+Arti." denotes applying articulation. Chamfer-$L_2 \times 10^{-4} m^2$, normal diff $\times 10^{-1}$.

	Global z+ AtlasNet [3]	Global z+ AtlasNet [3]+Arti.	Global z+ PCN [12]+Arti.	Pose params+ PCN [12]+Arti.	SCALE (Ours)
Chamfer-L_2	16.03	10.47	8.88	8.69	**8.41**
Normal diff	3.00	1.62	1.70	1.70	**1.32**

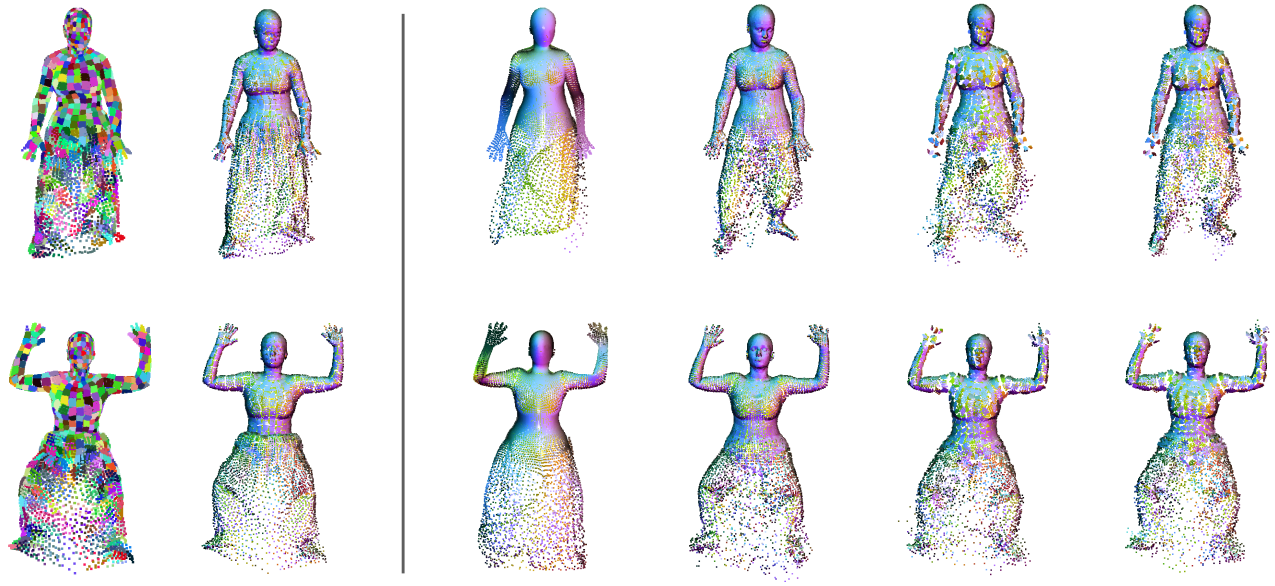

Figure S7: Qualitative results on the long dress data. "+Arti." denotes applying articulation.