Supplementary Material for Learning Implicit Templates for Point-Based Clothed Human Modeling

Siyou Lin, Hongwen Zhang, Zerong Zheng, Ruizhi Shao, and Yebin Liu

Tsinghua University, Beijing, China

A Implementation Details

A.1 Network Architectures

In the proposed FITE framework, four neural networks are employed for the learning of implicit template and pose-dependent deformation, including (i) the neural implicit template F^c in the canonical pose, (ii) the pose-agnostic template correction module C, (iii) U-Nets [13] U_d for projection-based pose encoding, and (iv) the pose-dependent deformation decoder D.

Implicit Templates For the implicit template F^c, we apply the same architecture as SNARF [2] as the shape network. Please refer to their original paper for details. The only modifications made to [2] are the replacement of the skinning network by our fixed diffused skinning (Section A.2) and the sampling strategy for training (Section A.4).

For the pose-agnostic template correction module C, we apply a small 4-layer MLP. Each layer except the last one of this MLP is followed by a 1D batch normalization [5] and a softplus [3] activation. The input dimension for C is $C_{\text{geom}} = 64$, and the intermediate layers' dimensions are $64, 64, 64, 3$.

Pose-dependent Deformations For the pose encoding U-Nets [13] U_d, we apply the same structure as the one used in POP [11] which downsamples the input 7 times, but with reduced channel sizes. Except the input, all other channels in the U-Nets are reduced to 1/4 the size of their counterparts in POP [11]. The output pose feature maps from each U_d thus have $C_{\text{pose}} = 16$ channels. Note that, with the reduced number of channels, even though we are using 4 U-Nets as opposed to a single U-Net in [11], the total number of parameters of our pose encoders is approximately 1/4 of that in [11]. Recall that we have $N_v = 4$ views for rendering position maps. With each view contributing $C_{\text{pose}} = 16$ channels, the number of channels for a concatenated pose feature vector is $N_v \cdot C_{\text{pose}} = 64$, which is consistent with [11] that extracts a 64-channel pose feature with a single network.

For the pose-dependent offset decoder D, we apply the same architecture as in [11]. The input is a concatenation of the pose feature ($N_v \cdot C_{\text{pose}} = 64$ channels), the per-point geometric feature ($C_{\text{geom}} = 64$ channels) and the coordinate of the base point in the canonical space p_k^c (3 channels), which add to 131 channels. The intermediate layers' dimensions are $256, 256, 256, 256, 387, 256, 256, 3$, where the 5th layer takes a skip connection from the input by concatenation. The 6th branches out with identical structures to predict offsets r_k^c and the normals n_k^c, respectively. All layers except the last are followed by batch normalizations [5] and softplus activations [3].

A.2 Computing Diffused Skinning

We compute diffused skinning in the whole 3D space based on the surface skinning weights of SMPL [9], which has 24 joints, and thus 24 components of skinning weights. Note that the zero pose (T-pose) of the SMPL body template is not suitable for computing volumetric skinning. We follow previous work [2,14,15] to define the canonical pose as standard T-pose with legs stretched out. Moreover, to avoid large distortions for long dresses, we set the Euler angle for both legs as 15 degrees. For fairness, baseline methods SNARF [2] and SCANimate [14] are trained with the same canonical pose.

Let $\{v_k^s\}_{k=1}^{N_s}$ be the vertices of the SMPL template T in this canonical pose with $N_s = 6890$, and let $w^s(v_k^s) = (w_1^s(v_k^s), \cdots, w_{24}^s(v_k^s)) \in \mathbb{R}^{24}$ be the SMPL skinning weights at v_k^s. The first step is to compute the along-surface gradients $\nabla_T w^s(v_k^s)$. Recall that in the 1D case, the gradient of a scalar function $y = f(x)$ can be approximated by

$$\nabla_x f(x) \approx \frac{f(x+h) - f(x)}{h} = \frac{f(x+h) - f(x)}{|h|} \cdot \frac{h}{|h|} \tag{1}$$

as long as h is small. In analogy, if $v_{k'}^s$ is a neighbor of v_k^s on the SMPL mesh template, i.e., connected by an edge, then

$$\frac{w(v_{k'}^s) - w(v_k^s)}{\|v_{k'}^s - v_k^s\|_2} \cdot \frac{v_{k'}^s - v_k^s}{\|v_{k'}^s - v_k^s\|_2} \tag{2}$$

approximates the gradient of w^s along the direction from v_k^s to $v_{k'}^s$. Note that Eq. (2) should be considered component-wise for w^s. To approximate $\nabla_T w^s(v_k^s)$, we can simply average Eq. (2) over all tangent directions emanating from v_k^s, i.e., if $\text{Nbr}(v_k^s)$ is the set of all neighboring vertices of v_k^s on T, then

$$\nabla_T w^s(v_k^s) \approx \frac{1}{|\text{Nbr}(v_k^s)|} \sum_{v \in \text{Nbr}(v_k^s)} \frac{w(v) - w(v_k^s)}{\|v - v_k^s\|_2} \cdot \frac{v - v_k^s}{\|v - v_k^s\|_2}. \tag{3}$$

We remark that in theory, the summation Eq. (3) should be weighted if the directions $v_k^s \to v$ are not evenly distributed. However, since the SMPL template has rather regular connectivity, we found that a simple average is enough.

With $\nabla_T w^s(v_k^s)$ already computed, we minimize the following energy as described in the main paper:

$$\lambda_{\text{p}}^s \int_{p \in T} \|w(p) - w^s(p)\|_2^2 + \lambda_{\text{g}}^s \int_{p \in T} \|\nabla_p w(p) - \nabla_T w^s(p)\|_2^2 + \lambda_{\text{reg}}^s \int_{\mathbb{R}^3} \left\|\nabla^2 w\right\|_2^2. \tag{4}$$

In a discrete setting, the first and the second term becomes summations:

$$\sum_{k=1}^{N_s} \|w(v_k^s) - w^s(v_k^s)\|_2^2 \quad \text{and} \quad \sum_{k=1}^{N_s} \|\nabla_p w(v_k^s) - \nabla_T w^s(v_k^s)\|_2^2 \,. \tag{5}$$

We apply an off-the-shelf solver [6] which applies the Galerkin formulation [8] to Eq. (4) and seeks a solution which is a linear combination of B-splines bases attached to the nodes of an octree. Note that the last regularization terms forces the solution to be linear (and thus also smooth). We set $\lambda_{\mathrm{p}}^s = 10^3$, $\lambda_{\mathrm{g}}^s = 5 \times 10^{-2}$, $\lambda_{\mathrm{reg}}^s = 1$ and set the maximum octree depth as 8 in the solver. The solution w is stored as a 256^3 grid. During the training of stage one, the forward skinning field is queried from this grid, instead of being jointly optimized with the canonical shape network.

A.3 Dataset Preprocessing

ReSynth The ReSynth [11] dataset contains 24 outfits, obtained by applying physics-based simulation [4] to a set of artist-designed outfits. This dataset contains non-closed point clouds which are unsuitable for implicit methods as well as our stage-one training. We thus apply screened Poisson surface reconstruction [7] to the dataset to obtain closed meshes. To avoid holes in the reconstructed meshes, we use the Dirichlet boundary condition in [7] (see Fig. S1 for a comparison of different boundary conditions). However, since simply resampling the reconstructed meshes for training FITE and POP lead to less details, we combine the original point samples with points newly sampled from the reconstruction according to the following rule: (i) For each specific scan, suppose $\mathcal{P}$ is the point set from the original ReSynth release (40k points per scan), and suppose $\hat{\mathcal{P}}$ is the point set newly sampled from reconstructed meshes (10k points per scan); (ii) Find all points $\hat{p} \in \hat{\mathcal{P}}$ such that $\|p - \hat{p}\|_2 \geq 0.01$ for all $p \in \mathcal{P}$; (iii) Randomly select a subset of $\mathcal{P}$ of the same size as the point set found in step (ii) and replace them with $\hat{p}$. For the long dress example (Felice 004), this would replace at most $\sim 2k$ points per scan, which does not affect the overall geometry. This processing step complements the original scans with points sampled from the reconstructed meshes. For fairness, POP [11] is also retrained on such data.

CAPE The CAPE [10] dataset contains registered scans for training, which do not requiring the same resampling strategy for ReSynth. We simply followed POP [10] to remove the global orientation and translation for training.

A.4 Training and Evaluation Details

For FITE, we train SNARF [2] with our diffused skinning for 8000 iterations per outfit in stage one. This takes $\sim$ 3.5 hours on a RTX 3090 GPU. During stage one, we also sample more densely near the hand to assure the appearance of finger within such a short training period. For stage two, we set $\lambda_{\mathrm{p}} = 10^4$, $\lambda_{\mathrm{n}} =$

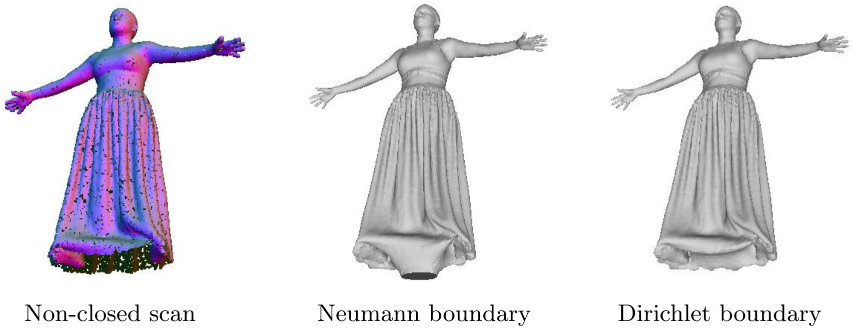

Fig. S1. Effects of different boundary conditions when applying screened Poisson reconstruction [7] to non-closed scans. We choose the Dirichlet boundary condition in data preprocessing to naturally close the holes.

1, $\lambda_{\mathrm{c,reg}} = 4 \times 10^2$, $\lambda_{\mathrm{r,reg}} = 2 \times 10^3$, $\lambda_{\mathrm{g,reg}} = 1$. Our model is trained for 400 epochs with a batch size of 4. Note that λ_{n} is set to 10^{-6} before the 250th epoch. This is to ensure that normal predictions are trained only after the points have stabilized. For all baselines we use the default settings in their official implementations for training.

For evaluation, we apply the Chamfer-L_2 distance d_{cham} and the cosine similarity to reconstructed meshes. Let M_1 and M_2 be the two meshes to compare, respectively. The metrics are obtained by first sampling two uniform point sets $\mathcal{P}_1$ and $\mathcal{P}_2$ and then computing:

$$d_{\mathrm{cham}} = \frac{1}{|\mathcal{P}_1|} \sum_{p \in \mathcal{P}_1} \|p - p'\|_2^2 + \frac{1}{|\mathcal{P}_2|} \sum_{q \in \mathcal{P}_2} \|q - q'\|_2^2 \,, \tag{6}$$

$$S_{\mathrm{cos}} = \frac{1}{|\mathcal{P}_1|} \left| \sum_{p \in \mathcal{P}_1} n_1(p) \cdot n_2(p') \right| + \frac{1}{|\mathcal{P}_2|} \left| \sum_{q \in \mathcal{P}_2} n_2(q) \cdot n_1(q') \right| , \tag{7}$$

where $p' \in \mathcal{P}_2$ is such that $\|q - q'\|_2$ is minimized and $q' \in \mathcal{P}_1$ is such that $\|q - q'\|_2$ is minimized.

A.5 User Study

The task of this user study is to determine the perceptual quality of FITE versus POP [11] in extrapolated poses with seen outfits. Each example shown to the viewer is side-by-side comparison of the outputs of FITE and POP with the same clothing in the same pose. The left-right order is randomly shuffled to prevent the preference to a certain side. The outfit and the pose for each example are randomly chosen with equal probability from the official ReSynth [11] test set. Moreover, the rendering direction (front or back) and the rendered geometry (point cloud or mesh) are also random. The viewers are asked to vote on the one with higher **overall** visual quality after considering factors such as realism, details and artifacts. Fig. S2 shows a few examples of the side-by-side image pairs used in this study. The votes are counted per example, i.e., if a viewer votes FITE as having higher visual quality for a specific image pair, we add one vote to FITE, otherwise the vote goes to POP.

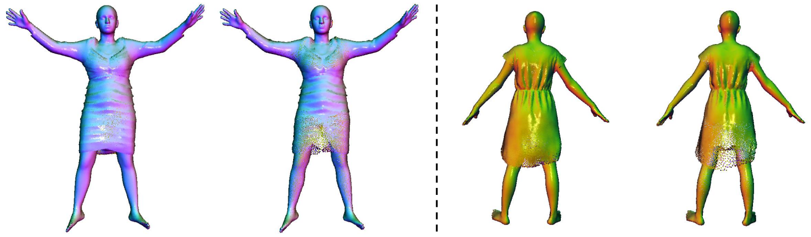

Fig. S2. Examples of image pairs used in the user study (rendered with Open3D [16]).

There are 421 individuals participating in the user study, where each is given 20 image pairs of either point clouds or meshes. The user study receives a total of 4690 votes on the point cloud results, among which 3537 (75.42%) favor FITE, and a total of 3730 votes on the meshed results, among which 2808 (59.37%) favor FITE.

A.6 Novel Scan Animation

Given one or a few novel scans, we first train a canonical template using these novel scans with diffused skinning. After obtaining this canonical template, we uniformly sample a set of points $\{p_k^c\}_k$ in the template, along with the corresponding skinning weights. Let $\{p_{k,i}^c\}_k$ denote the point set sampled on the i-th canonical template used in pretraining a FITE model. We then initialize the geometric features for $\{p_k^c\}$ as follows: (i) For each subject i used in pretraining, propagate the pretrained geometric feature of $\{p_{k,i}^c\}_k$ to $\{p_k^c\}_k$ via kNN ($k = 16$) and inverse distance weighting; (ii) Average these geometric features over i. After this initialization, we forward the model and minimize the difference between the given scans and the predicted point clouds (only the geometric features are optimized and network weights are fixed). After convergence, the geometric features and the pretrained network can be used to animate the scan to novel poses.

B Extended Evaluations

In this section we provided extended evaluations, including: (i) more results of interpolation and extrapolation experiments; (ii) ablation studies that evaluate the individual modules.

B.1 More Results of Interpolation and Extrapolation Experiments

Table S1 and Table S2 show the complete quantitative results of interpolation experiments on the ReSynth dataset [11] and the CAPE dataset [10], respectively. Since implicit methods have been shown to perform notably worse (see Table 1 and Fig. 4 in the main paper), we only evaluate the cross-outfit models of POP [11] and FITE. Our method exhibits clear advantages on ReSynth where the outfits vary in topology and style. On the other hand, for the CAPE dataset which contains only tight clothing, our method does not perform notably better since using learned templates instead of SMPL/SMPL-X [9,12] is not necessary. However, improving metrics for tight clothing is not the major focus of this work.

We report qualitative evaluations for pose extrapolation experiments in Table S3 and Table S4. These models are trained and tested on the official ReSynth train/test split. However, since extrapolated poses may correspond to stochastric clothing appearances, these metrics are only included for completeness and may not faithfully reflect the modeling capability of different methods. In Fig. S3, S4 and S5, we present mode qualitative results evaluated on extrapolated poses. The advantages of our method over POP [11] is obvious: no seams in clothing, more uniformly distributed points and better details.

Table S1. Quantitative results of interpolation experiments on the ReSynth dataset. Note that all d_{cham} reported below have been multiplied by 10^5.

Method	ReSynth Data							
	Alexandra 006		Anna 001		Beatrice 025		Carla 006	
	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)
POP	0.550	0.950	0.557	0.959	0.203	0.963	0.154	0.968
FITE	0.449	0.958	0.385	0.965	0.251	0.965	0.169	0.971
	Celina 005		Cindy 005		Corey 006		Eric 006	
	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)
POP	0.175	0.970	0.212	0.967	0.295	0.960	0.521	0.945
FITE	0.193	0.973	0.232	0.970	0.277	0.965	0.445	0.950
	Eric 035		Carla 004		Christine 027		Felice 004	
	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)	d_{cham} ($\downarrow$)	$S_{\cos}$ ($\uparrow$)
POP	1.625	0.894	0.485	0.939	0.421	0.960	1.718	0.920
FITE	0.615	0.923	0.299	0.956	0.455	0.963	1.355	0.933

B.2 Ablation Studies

We conduct ablation studies to evaluate the key components of our method: projection-based pose encoding (PPE) and diffused skinning. The experiments in this section are conducted under the same settings as the interpolation experiments.

Table S2. Quantitative results of interpolation experiments on the CAPE dataset. Note that all d_{cham} reported below have been multiplied by 10^7.

	CAPE Data							
Method	00096 jerseyshort		00096 longshort		00096 shirtlong		00096 shirtshort	
	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)
POP	4.372	0.986	3.546	0.988	6.687	0.981	6.557	0.981
FITE	4.168	0.987	5.335	0.987	8.571	0.981	6.653	0.982
	00096 shortlong		00096 shortshort		00215 jerseyshort		00215 longshort	
	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)
POP	5.994	0.982	4.836	0.985	6.626	0.983	6.087	0.984
FITE	5.892	0.982	4.813	0.986	6.856	0.983	6.131	0.985
	00215 poloshort		00215 shortlong		03375 blazerlong		03375 longlong	
	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)
POP	9.099	0.978	11.038	0.977	25.227	0.966	19.389	0.971
FITE	7.569	0.980	11.387	0.978	27.409	0.964	20.280	0.971
	03375 shortlong		03375 shortshort					
	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)				
POP	14.968	0.976	16.812	0.975				
FITE	16.207	0.976	17.661	0.975				

Table S3. Quantitative results of extrapolation experiments on the ReSynth dataset. Note that all d_{cham} reported below have been multiplied by 10^5.

	ReSynth Data							
Method	Alexandra 006		Anna 001		Beatrice 025		Carla 006	
	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)
POP	2.318	0.899	0.972	0.941	0.392	0.956	0.413	0.952
FITE	2.248	0.903	1.105	0.939	0.525	0.954	0.492	0.948
	Celina 005		Cindy 005		Corey 006		Eric 006	
	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)
POP	0.468	0.954	0.619	0.946	0.716	0.935	1.100	0.920
FITE	0.574	0.953	0.704	0.944	0.829	0.932	1.133	0.918
	Eric 035		Carla 004		Christine 027		Felice 004	
	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)	d_{cham} (↓)	S_{cos} (↑)
POP	2.761	0.864	0.796	0.926	3.021	0.914	16.768	0.763
FITE	2.071	0.870	0.754	0.936	3.033	0.913	16.106	0.759

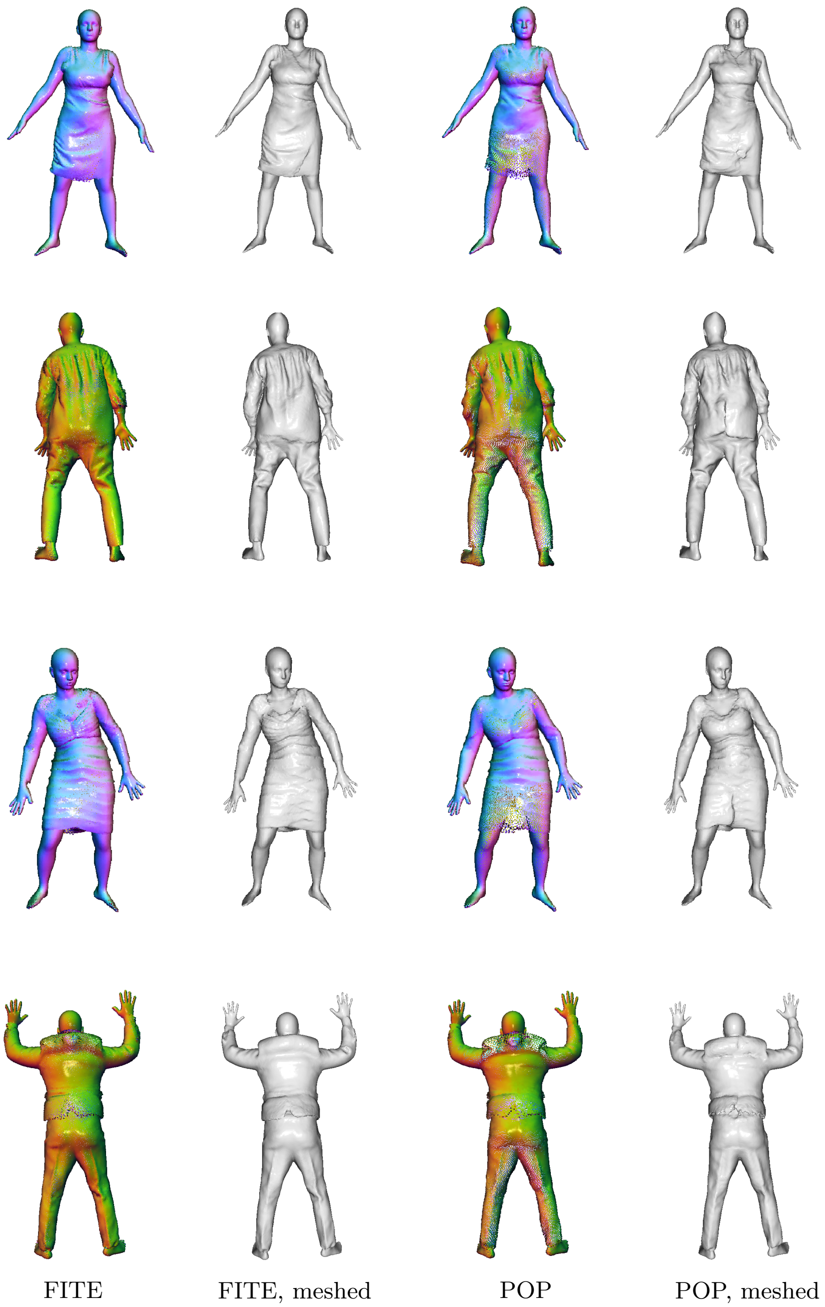

Fig. S3. Extended qualitative results for pose extrapolation experiments.

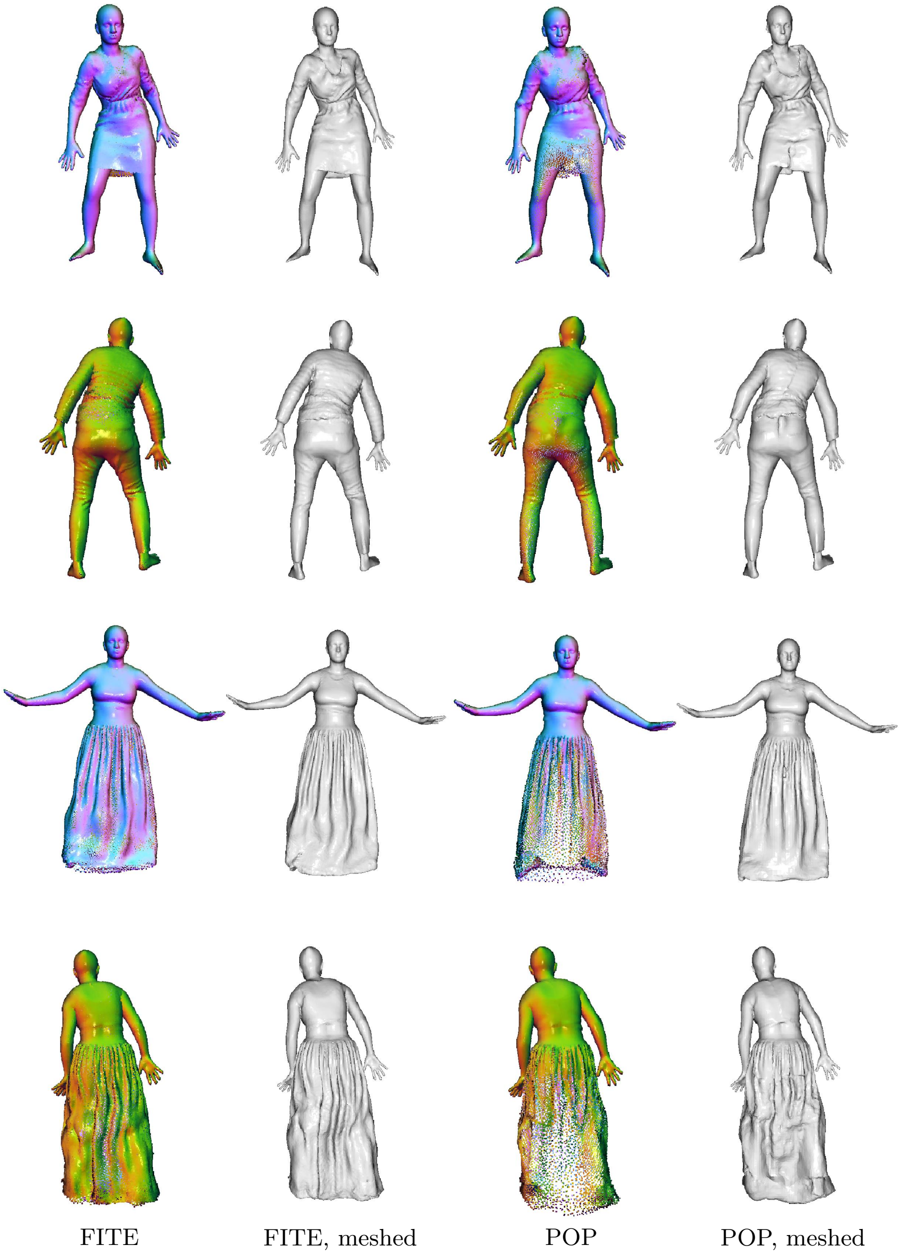

Fig. S4. Extended qualitative results for pose extrapolation experiments.

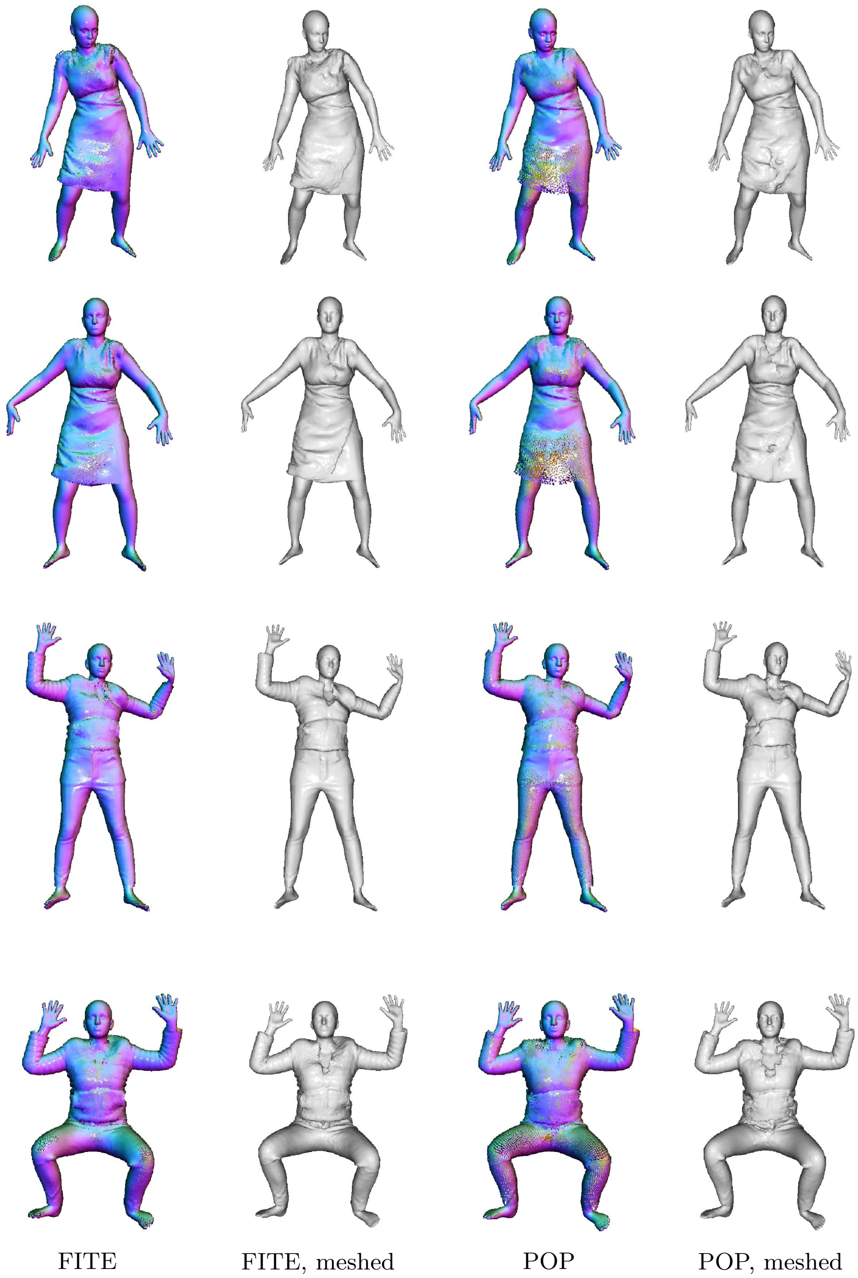

Fig. S5. Extended qualitative results for pose extrapolation experiments.

Table S4. Quantitative results of extrapolation experiments on the CAPE dataset. Note that all d_{cham} reported below have been multiplied by 10^5.

Method	CAPE Data							
	00096 jerseyshort		00096 longshort		00096 shirtlong		00096 shirtshort	
	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$
POP	0.450	0.970	0.414	0.974	0.559	0.958	0.514	0.960
FITE	0.722	0.951	0.706	0.953	0.850	0.942	0.827	0.941
	00096 shortlong		00096 shortshort		00215 jerseyshort		00215 longshort	
	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$
POP	0.700	0.953	0.475	0.968	0.413	0.968	0.408	0.969
FITE	1.000	0.935	0.787	0.948	0.695	0.956	0.674	0.956
	00215 poloshort		00215 shortlong		03375 blazerlong		03375 longlong	
	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$
POP	0.598	0.958	0.780	0.946	0.776	0.945	0.740	0.946
FITE	0.805	0.946	0.975	0.936	1.361	0.924	1.259	0.926
	03375 shortlong		03375 shortshort					
	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$				
POP	0.522	0.958	0.523	0.956				
FITE	0.975	0.937	1.068	0.935				

Projection-based pose encoding (PPE) Since UV-based pose encoding in POP [11] cannot be applied to the learned templates, we evaluate PPE by replacing UV encoding in POP [11] with PPE (named as POP+PPE). The results are shown in Table S5. We can observe a drop in the metrics after using PPE on POP directly. This indicates PPE alone cannot outperform UV encoding in terms of modeling accuracy. However, PPE is still advantageous in two aspects: (i) It solves the discontinuity problem in UV maps (Fig. S6); (ii) It is an indispensable component of our framework, since the learned templates do not have a common UV mapping and UV encoding is not applicable. Moreover, note that POP+PPE is the same as FITE minus learned templates. This study is also a demonstration of the benefit of using learned templates instead of minimal body models.

Diffused skinning We evaluate our diffused skinning by replacing it with learned skinning used in SNARF [2] and the nearest-SMPL-point skinning in LoopReg [1]. We name these two variants as FITE-LS and FITE-NS, respectively. The quantitative results are reported in Table S5. FITE-LS performs worse in terms of metrics, and may fail due to local minima where an incorrect template and an incorrect skinning field together form an *accidentally correct* posed geometry (Fig. S7). On the other hand, FITE-NS can also obtain good templates and achieve comparable performance to FITE. However, for loose clothing on which the discontinuity of nearest-point skinning manifests, qualitative artifacts can be observed, especially in extrapolated poses (Fig. S8).

Table S5. Quantitative results of ablation studies on the ReSynth dataset. Note that all d_{cham} reported below have been multiplied by 10^5.

Method	ReSynth Data							
	Alexandra 006		Anna 001		Beatrice 025		Carla 006	
	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$
POP	0.550	0.950	0.557	0.959	**0.203**	0.963	**0.154**	**0.968**
POP+PPE	0.603	0.952	0.914	0.956	0.275	0.962	0.204	0.969
FITE-LS	1.593	0.940	0.660	0.954	0.768	0.953	2.508	0.923
FITE-NS	0.470	**0.958**	0.435	0.964	0.261	**0.965**	0.165	0.971
FITE	**0.449**	**0.958**	**0.385**	**0.965**	0.251	**0.965**	0.169	0.971
	Celina 005		Cindy 005		Corey 006		Eric 006	
	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$
POP	**0.175**	0.970	**0.212**	0.967	0.295	0.960	0.521	0.945
POP+PPE	0.241	0.970	0.320	0.967	0.362	0.961	0.637	0.944
FITE-LS	0.299	0.968	0.631	0.961	0.937	0.952	0.879	0.938
FITE-NS	0.194	**0.973**	0.243	**0.970**	0.278	0.964	0.455	**0.950**
FITE	0.193	**0.973**	0.232	**0.970**	**0.277**	**0.965**	**0.445**	**0.950**
	Eric 035		Carla 004		Christine 027		Felice 004	
	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$	d_{cham} $(\downarrow)$	S_{cos} $(\uparrow)$
POP	1.625	0.894	0.485	0.939	**0.421**	0.960	1.718	0.920
POP+PPE	1.707	0.898	0.537	0.943	0.560	0.959	2.298	0.912
FITE-LS	0.771	0.913	1.205	0.937	0.646	0.957	1.903	0.923
FITE-NS	0.621	0.922	0.300	**0.956**	0.488	**0.963**	1.541	0.926
FITE	**0.615**	**0.923**	**0.299**	**0.956**	0.455	**0.963**	**1.355**	**0.933**

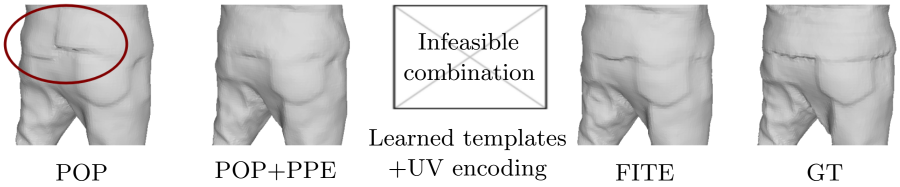

Fig. S6. Compared with UV encoding in POP [11], our projection-based pose encoding (PPE) can solve the discontinuity problem and can encode learned templates without common a UV map.

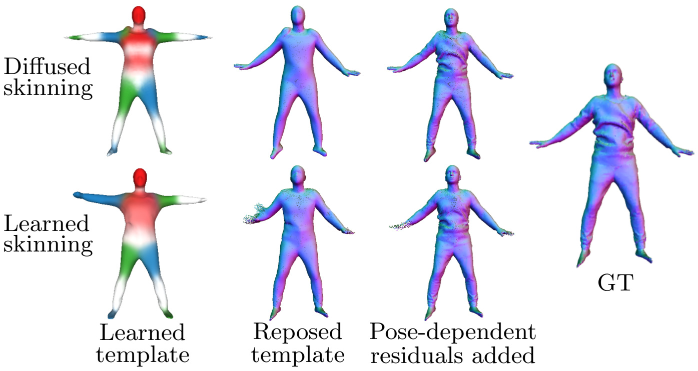

Fig. S7. Visualization of diffused skinning and learned skinning. Learned skinning may fail due to local minima, leading to an incorrect reposed geometry.

C Limitations

Our work decomposes clothing deformations into articulated motions of a clothed template (stage one) and pose-dependent non-rigid deformations (stage two). However, LBS cannot fully account for clothing deformations, especially for loose clothing. This means a poorly posed template possibly does not help the generation of details in stage two. In particular, if the LBS-posed template does not match the ground truth point cloud very well, then the Chamfer loss may induce nonbalanced correspondences between the ground truth and the template to deform. For certain extreme poses, this may cause the output point cloud to become highly nonuniformly distributed, leading to suboptimal performance (Fig. S9). Moreover, LBS is in general not physically realistic for loose clothing. Hence, generalizing to novel poses outside the training distribution may lead to physically absurd results (Fig. S10). With our coarse-to-fine two-stage idea, future work may explore replacing LBS in stage one with coarse-level physics-based simulation, which may possibly improve the performance for certain outfits.

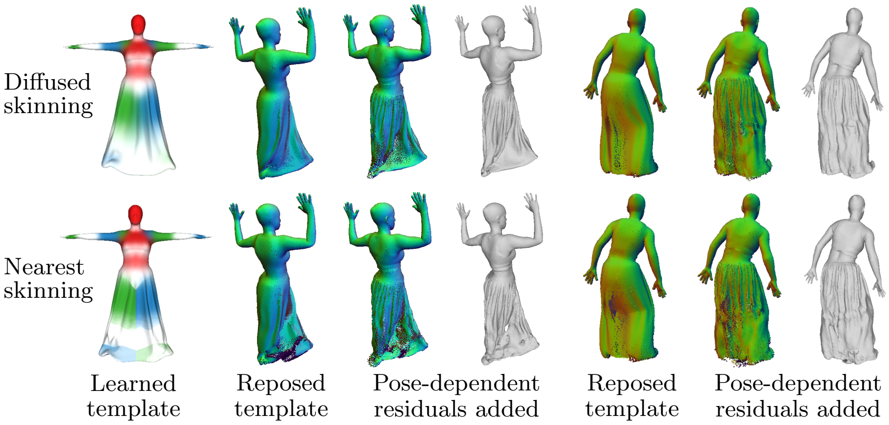

Fig. S8. Visualization of diffused skinning and nearest-point skinning. Nearest-point skinning becomes discontinuous for loose clothing and leads to poor geometry.

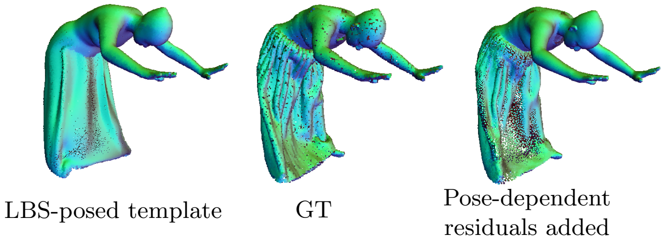

Fig. S9. Failure case 1: The Chamfer loss does not always induce good correspondences between GT and the LBS-posed template, leading to suboptimal performance.

References

1. Bhatnagar, B.L., Sminchisescu, C., Theobalt, C., Pons-Moll, G.: Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. In: Advances in Neural Information Processing Systems (NIPS) (December 2020)
2. Chen, X., Zheng, Y., Black, M.J., Hilliges, O., Geiger, A.: Snarf: Differentiable forward skinning for animating non-rigid neural implicit shapes. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 11594–11604 (October 2021)
3. Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., Garcia, R.: Incorporating second-order functional knowledge for better option pricing. In: Proceedings of the 13th International Conference on Neural Information Processing Systems. p. 451–457. NIPS'00, MIT Press, Cambridge, MA, USA (2000)
4. Guan, P., Reiss, L., Hirshberg, D.A., Weiss, A., Black, M.J.: DRAPE: DRessing Any PErson. ACM Transactions on Graphics (TOG) **31**(4), 35–1 (2012)
5. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Confer-

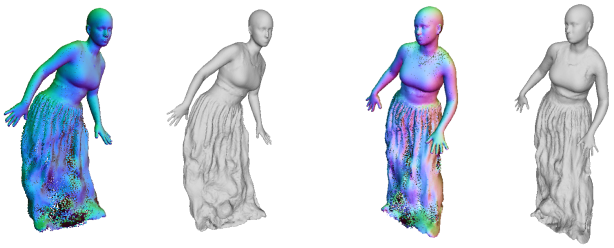

Fig. S10. Failure case 2: Generalizing to poses outside the training distribution may not be physically realistic.

ence on International Conference on Machine Learning (ICML). ICML'15, vol. 37, p. 448–456. JMLR.org (2015)

6. Kazhdan, M.: Pointinterpolant. https://github.com/mkazhdan/PoissonRecon (2021)
7. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. ACM Trans. Graph. **32**(3) (Jul 2013). https://doi.org/10.1145/2487228.2487237, https://doi.org/10.1145/2487228.2487237
8. Kazhdan, M., Hoppe, H.: An adaptive multi-grid solver for applications in computer graphics. Computer Graphics Forum **38**(1), 138–150 (2019). https://doi.org/https://doi.org/10.1111/cgf.13449, https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13449
9. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: Smpl: A skinned multi-person linear model. ACM Transactions on Graphics (TOG) **34**(6), 248 (2015)
10. Ma, Q., Yang, J., Ranjan, A., Pujades, S., Pons-Moll, G., Tang, S., Black, M.J.: Learning to dress 3d people in generative clothing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6469–6478 (2020)
11. Ma, Q., Yang, J., Tang, S., Black, M.J.: The power of points for modeling humans in clothing. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (Oct 2021)
12. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3d hands, face, and body from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10975–10985 (2019)
13. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention. pp. 234–241. Springer (2015)
14. Saito, S., Yang, J., Ma, Q., Black, M.J.: Scanimate: Weakly supervised learning of skinned clothed avatar networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2886–2897 (2021)
15. Wang, S., Mihajlovic, M., Ma, Q., Geiger, A., Tang, S.: Metaavatar: Learning animatable clothed human models from few depth images. In: Advances in Neural Information Processing Systems (NIPS) (2021)
16. Zhou, Q.Y., Park, J., Koltun, V.: Open3D: A modern library for 3D data processing. arXiv:1801.09847 (2018)