Supplementary Material for ISP: Multi-Layered Garment Draping with Implicit Sewing Patterns

Anonymous Author(s)
Affiliation
Address
email

1 Additional Results

1.1 Texture Editing

As illustrated in Fig. 1, with our ISP, the texture of the garment can be easily edited by drawing on the UV panels. The figures drawn on the panel of one garment can be directly transferred to others as shown by Fig. 1(c), since panels are defined on the same UV space.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

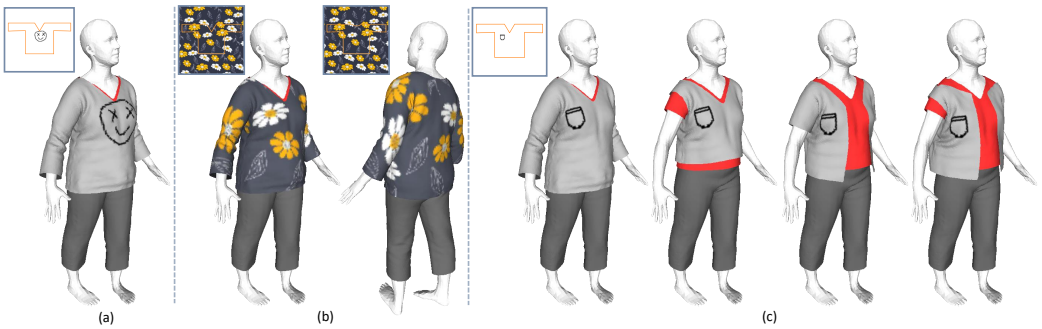

Figure 1: **Texture editing. (a)** A smiling face drawn on the front panel. **(b)** A flower pattern painted on both the front and the back panels. **(c)** A pocket drawn for the left shirt can be transferred to different garments.

1.2 Garment Reconstruction

1.2.1 Evaluation Results

Train	CHD ($\times 10^{-4}$, $\downarrow$)	NC (%, $\uparrow$)	Time (ms, $\downarrow$)
UDF - 128	0.714	98.53	690
UDF - 256	0.408	98.93	2784
UDF - 512	0.331	98.43	17362
Ours - 128	0.538	98.80	**25**
Ours - 256	0.321	99.08	75
Ours - 512	**0.267**	**99.14**	262

Test	CHD ($\times 10^{-4}$, $\downarrow$)	NC (%, $\uparrow$)
UDF - 128	0.734	97.79
UDF - 256	0.403	98.64
UDF - 512	0.324	98.27
Ours - 128	0.583	98.62
Ours - 256	0.362	98.85
Ours - 512	**0.304**	**98.89**

Table 1: Comparison of our method to UDF on skirts under the resolutions of 128, 256 and 512.

Train	CHD ($\times 10^{-4}$, $\downarrow$)	NC (%, $\uparrow$)	Time (ms, $\downarrow$)
UDF - 128	0.758	98.22	661
UDF - 256	0.430	98.56	2650
UDF - 512	0.342	98.33	17141
Ours - 128	0.545	98.28	**25**
Ours - 256	0.363	98.59	82
Ours - 512	**0.317**	**98.62**	269

Test	CHD ($\times 10^{-4}$, $\downarrow$)	NC (%, $\uparrow$)
UDF - 128	0.752	97.41
UDF - 256	0.425	98.09
UDF - 512	0.350	97.63
Ours - 128	0.529	98.03
Ours - 256	0.346	98.31
Ours - 512	**0.300**	**98.32**

Table 2: Comparison of our method to UDF on trousers under the resolutions of 128, 256 and 512.

In Tabs. 1 and 2 we report the reconstruction results for skirts and trousers on the training and the test set. Similar to the results on shirts shown in the main paper, our method achieves better reconstruction quality than UDF [1] with lower CHD and higher NC at all resolutions, and needs less time to reconstruct a single mesh. Figs. 2 to 4 show the qualitative results reconstructed by our method for shirts, skirts and trousers respectively.

1.2.2 Latent Space Interpolation

In Figs. 5 to 7, we display the results of interpolation in the latent space of shirts, skirts and trousers respectively. We observe a smooth transformation in both the reconstructed sewing patterns and the garment meshes, despite the different topology and geometry of the given garments.

1.2.3 Comparison with AtlasNet

In Fig. 8, we compare our method with AtlasNet [2] which learns to deform a square patch. AtlasNet struggles to learn a mapping function capable of accurately deforming the square patch to produce a surface that matches the ground truth, especially in the collar region. In contrast, our method leverages the pattern parameterization network $\mathcal{I}_\Theta$ to simplify the training of our mapping function $\mathcal{A}_\Phi$. Specifically, our approach only requires learning the mapping for points within the panels, resulting in a reconstruction that is more faithful to the ground truth.

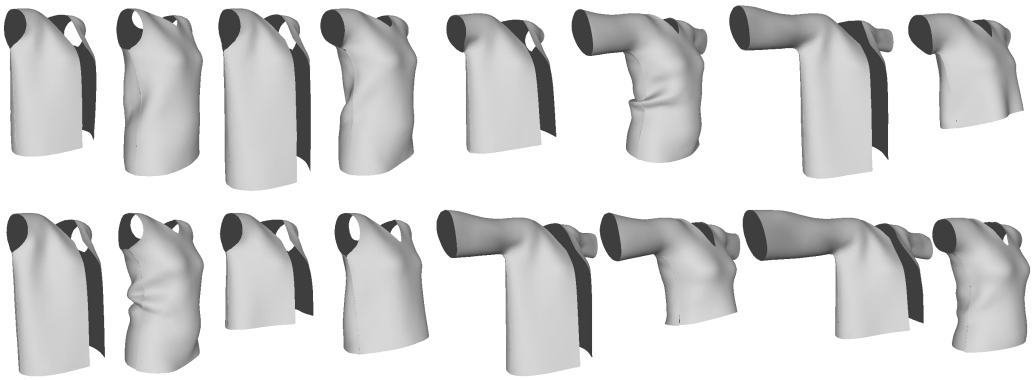

Figure 2: Reconstruction samples of shirts.

Figure 3: Reconstruction samples of skirts.

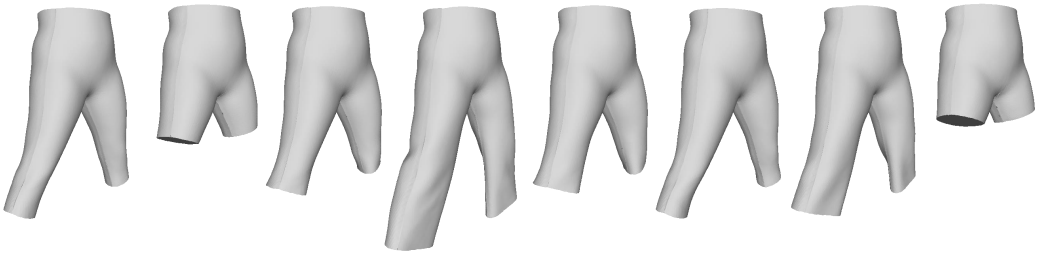

Figure 4: Reconstruction samples of trousers.

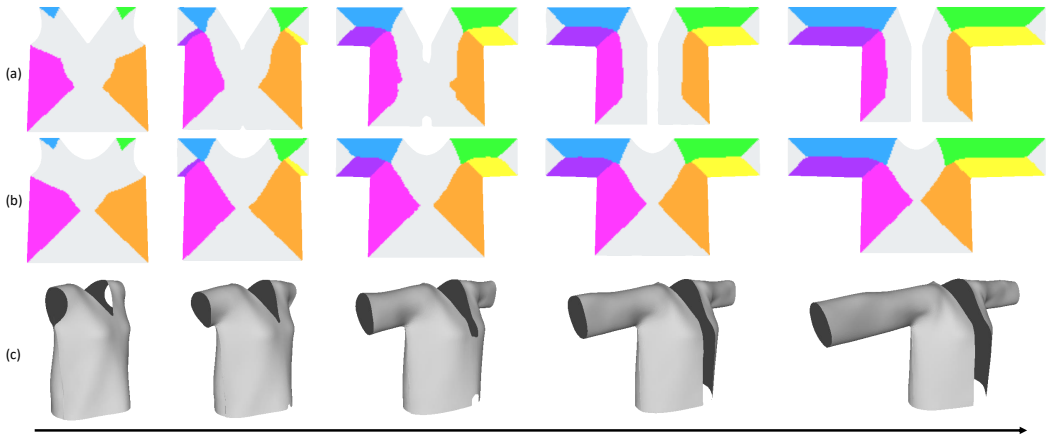

Figure 5: **Interpolation**. We interpolate the latent code from a sleeveless shirt to a long-sleeve jacket. (a) and (b) show the reconstructed front and back panels, where the colors on them denote the edge label fields. (c) shows the reconstructed mesh.

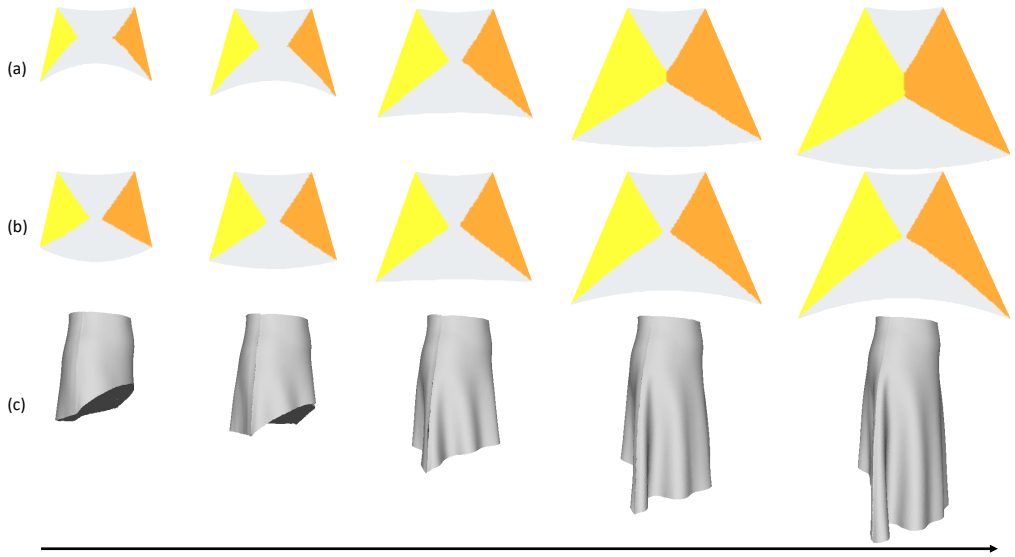

Figure 6: **Interpolation**. We interpolate the latent code from a short tight skirt to a long loose skirt. (a) and (b) show the reconstructed front and back panels, where the colors on them denote the edge label fields. (c) shows the reconstructed mesh.

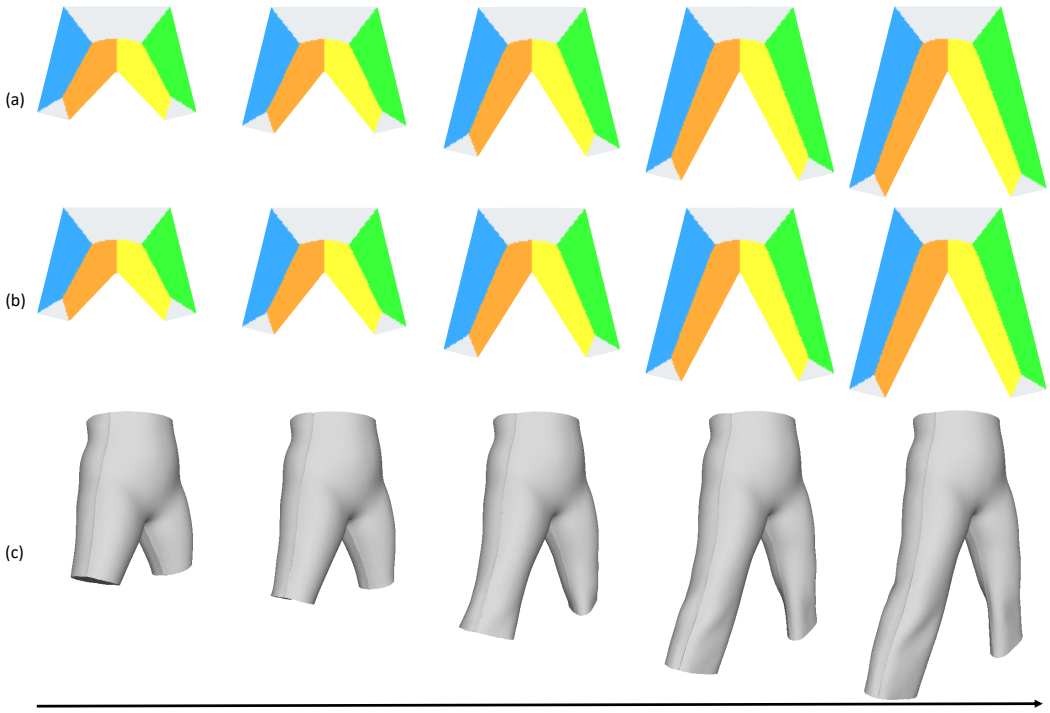

Figure 7: **Interpolation**. We interpolate the latent code from a pair of short trousers to a pair of long trousers. (a) and (b) show the reconstructed front and back panels, where the colors on them denote the edge label fields. (c) shows the reconstructed mesh.

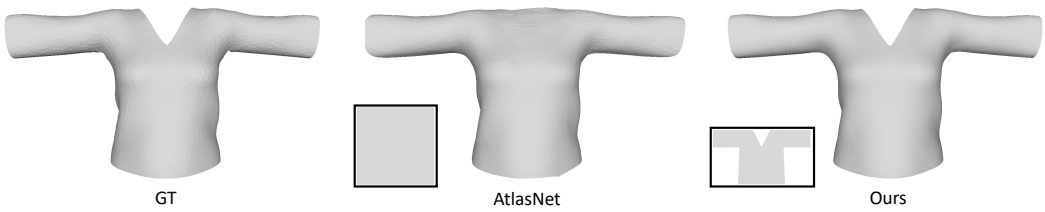

Figure 8: **Comparison with AtlasNet**. The meshes in black boxes are the source patches where the mapping function is applied: a square patch for AtlastNet, a garment panel for our method.

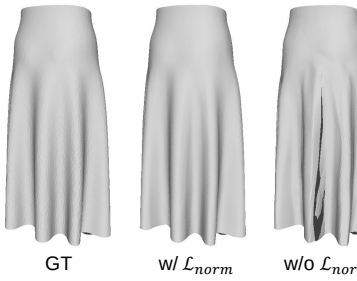

Figure 9: **Ablation on** $\mathcal{L}_{normal}$. Training without it yields flipped triangle faces (shown in dark grey).

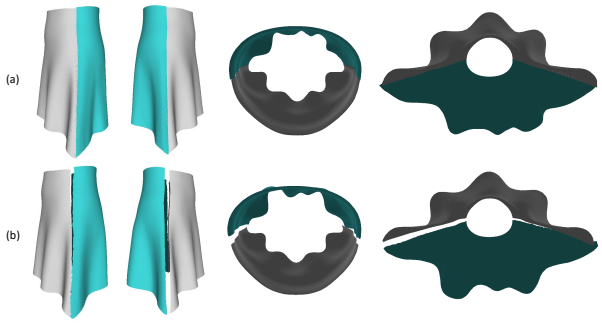

Figure 10: **Ablation on** $\mathcal{L}_{consist}$. The skirts reconstructed by the models trained (a) with $\mathcal{L}_{consist}$ and (b) without $\mathcal{L}_{consist}$. The latter has a gap between its front and back panels.

1.2.4 Ablation Study

To investigate the impact of the loss terms of Eq. (4) in the main paper on reconstruction quality, we performed an ablation study. We report the results of reconstructing 300 skirts using models trained with and without $\mathcal{L}_{normal}$ in Tab. 3. We observe that training with $\mathcal{L}_{normal}$ reduces the CHD and increases the NC, thus improving the reconstruction accuracy. In contrast, without $\mathcal{L}_{normal}$, the model fails to learn the correct parameterization for the garment, resulting in a mesh with inverted faces as illustrated in Fig. 9. Fig. 10 shows a comparison of the results obtained by training models without and with $\mathcal{L}_{consist}$ to help stitch the front and back panels. We can notice that without $\mathcal{L}_{consist}$, a spatial gap exists between the front (in gray) and back (in cyan) surfaces.

1.3 Garment Draping

Fig. 11 and Fig. 12(a) present additional comparisons of draping results for our method, DIG [3] and DrapeNet [1]. Our results show higher fidelity and fewer artifacts compared to the other methods. We also compare our method with SNUG [4], a self-supervised method that relies on mesh templates for garment representation and trains one network for each clothing item. In Fig. 12(b), we show the qualitative results on the same shirt. Our results are either comparative or visually superior to those of SNUG, despite using a single draping network for a whole garment category.

	CHD ($\times 10^{-4}$, $\downarrow$)	NC (%, $\uparrow$)
w/o $\mathcal{L}_{normal}$	0.324	98.69
w/ $\mathcal{L}_{normal}$	0.321	99.08

Table 3: Comparison of the results reconstructed by the models trained w/o and w/ $\mathcal{L}_{normal}$.

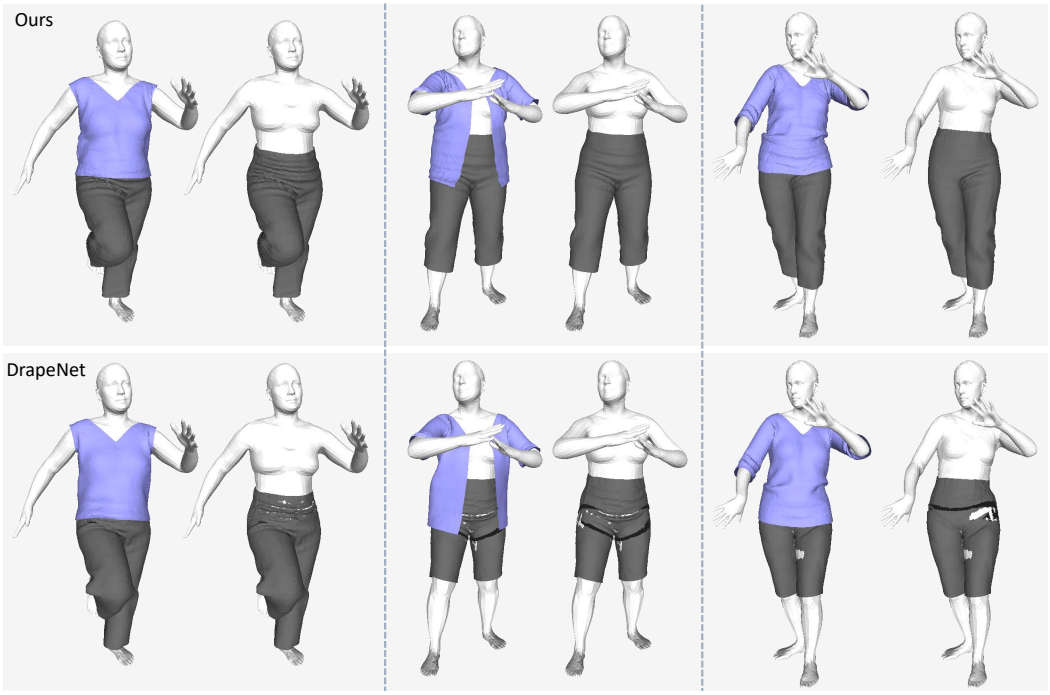

Figure 11: The comparison of draping results of our method and DrapeNet.

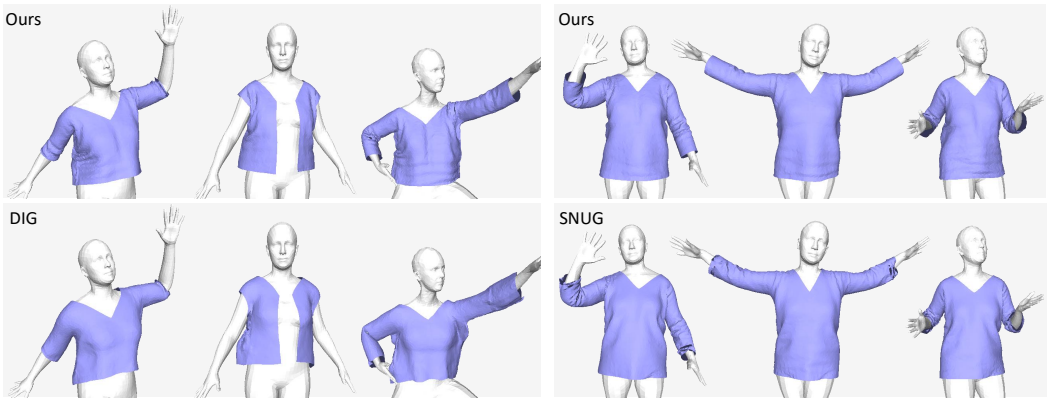

Figure 12: The comparison of draping results for (a) our method vs. DIG, and (b) our method vs. SNUG.

2 Technical Details

2.1 Sewing Patterns for Trousers and Skirts

In Fig. 13, we show the sewing patterns used in our experiments for trousers and skirts.

2.2 Mesh Triangulation

In this section, we detail the meshing process of ISP. We first create a square 2D mesh $\mathbf{T}$ for Ω as shown on the left of Fig. 14. Given the latent code $\mathbf{z}$ of a specific garment, for each vertex $v \in V_\Omega$, we compute its signed distance value s and edge label c with $(s, c) = \mathcal{I}_\Theta(v, \mathbf{z})$. The 2D front and back panel meshes $\mathbf{T}_{P_f}$ and $\mathbf{T}_{P_b}$ are constructed by keeping vertices of $\mathbf{T}$ with negative signed distance (the blue region of the colored gird in Fig. 14) and those that have positive signed distance but belong to the edges crossing the 0 iso-level (the gray region of the colored gird in Fig. 14). For the later ones,

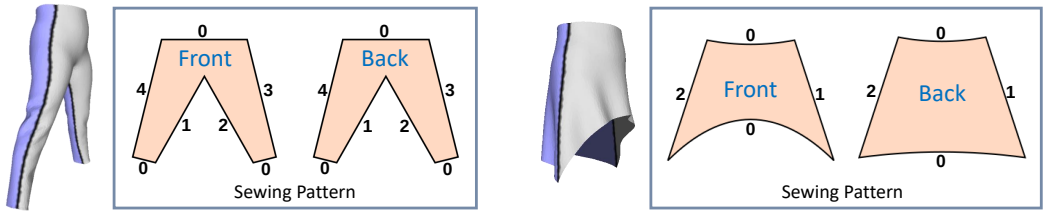

Figure 13: The sewing patterns for (a) trousers and (b) skirts. The front mesh surfaces are in gray and the back ones in blue.

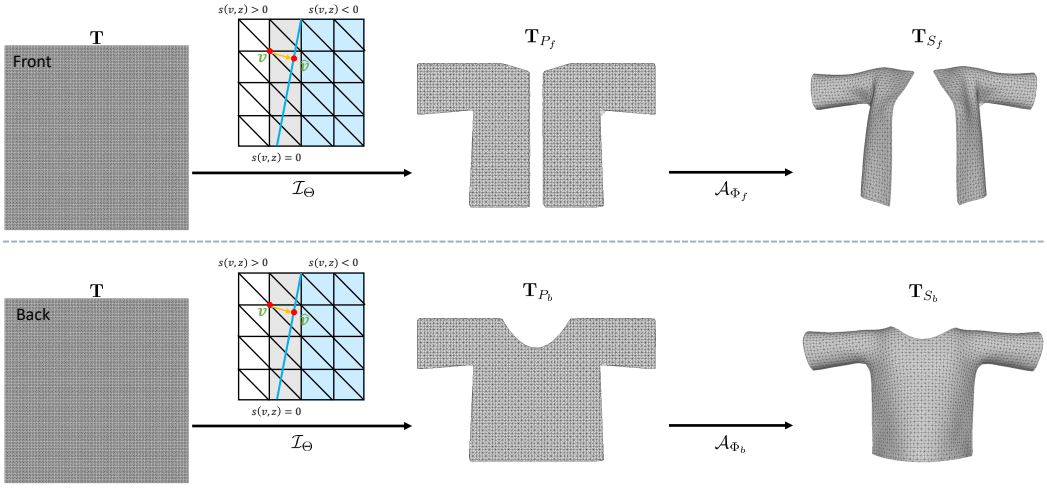

Figure 14: **Meshing Process.** Starting with a square 2D mesh $\mathbf{T}$, we first extract the front and back panel meshes $\mathbf{T}_{P_f}$ and $\mathbf{T}_{P_b}$ using the implicit function $\mathcal{I}_\Theta$. Then we lift $\mathbf{T}_{P_f}$ and $\mathbf{T}_{P_b}$ to 3D to get the surface meshes $\mathbf{T}_{S_f}$ and $\mathbf{T}_{S_b}$ by querying $\mathcal{A}_{\Phi_f}$ and $\mathcal{A}_{\Phi_b}$ on their vertices respectively.

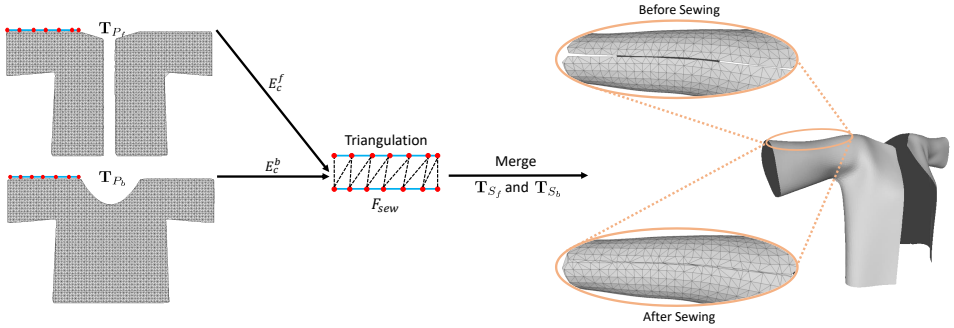

Figure 15: **Sewing process.** Left: the boundary vertices belonging to the sewing edges E_c^f and E_c^b are marked in red. Middle: The triangulation is performed between E_c^f and E_c^b to create new faces F_{sew}. Right: The mesh generated by merging the front and back surfaces $\mathbf{T}_{S_f}$ and $\mathbf{T}_{S_b}$ with F_{sew}.

we adjust their positions from v to $\hat{v} = v - s(v, \mathbf{z})\nabla s(v, \mathbf{z})$ to project them to the zero level set (the blue line). Finally, we query $\mathcal{A}_{\Phi_f}$ and $\mathcal{A}_{\Phi_b}$ on each vertex of $\mathbf{T}_{P_f}$ and $\mathbf{T}_{P_b}$ respectively to lift them to 3D, giving us the front and back surfaces $\mathbf{T}_{S_f}$ and $\mathbf{T}_{S_b}$ as shown on the right of Fig. 14.

To sew the lifted front and back surfaces $\mathbf{T}_{S_f}$ and $\mathbf{T}_{S_b}$, we perform triangulation with the help of panel meshes $\mathbf{T}_{P_f}$ and $\mathbf{T}_{P_b}$. As illustrated in Fig. 15, we group the boundary vertices of $\mathbf{T}_{P_f}$ and $\mathbf{T}_{P_b}$ whose predicted labels are the same (c, with $c > 0$) to form the sewing edges E_c^f and E_c^b for the front and back panels separately. Then we create faces F_{sew} between the vertices of E_c^f and E_c^b, and use F_{sew} to merge the meshes of $\mathbf{T}_{S_f}$ and $\mathbf{T}_{S_b}$, which gives us the final assembled garment mesh.

2.3 Proof of the Differentiability of ISP

According to the *Theorem 1* of [5], for an SDF s and the point $\mathbf{x}_0$ lying on the 0 iso-level $l = \{\mathbf{q}|s(\mathbf{q}, \mathbf{z}) = 0, \mathbf{q} \in \Omega\}$, we have

$$\frac{\partial \mathbf{x}_0}{\partial s} = -\nabla s(\mathbf{x}_0, \mathbf{z}). \tag{1}$$

For point $\mathbf{x}$ lying on the α iso-level, we can have $s(\mathbf{x}, \mathbf{z}) = \alpha$, where α is a constant. Let $s_\alpha = s - \alpha$, then $\mathbf{x}$ lies on the 0 iso-level of s_α. Based on Eq. 1, we can have

$$\frac{\partial \mathbf{x}}{\partial s_\alpha} = -\nabla s_\alpha(\mathbf{x}, \mathbf{z}) = -\nabla s(\mathbf{x}, \mathbf{z}). \tag{2}$$

Assume v is a vertex of the mesh $\mathbf{T}_G$ reconstructed by ISP and $\mathbf{x}$ is the point on the UV space that satisfies $v = \mathcal{A}_\Phi(\mathbf{x}, \mathbf{z})$, then it holds that

$$\frac{\partial v}{\partial \mathbf{z}} = \frac{\partial \mathcal{A}_\Phi}{\partial \mathbf{z}}(\mathbf{x}, \mathbf{z}) + \frac{\partial \mathcal{A}_\Phi}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \mathbf{z}}(\mathbf{x}, \mathbf{z}), \tag{3}$$

$$= \frac{\partial \mathcal{A}_\Phi}{\partial \mathbf{z}}(\mathbf{x}, \mathbf{z}) + \frac{\partial \mathcal{A}_\Phi}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial s_\alpha} \frac{\partial s_\alpha}{\partial s} \frac{\partial s}{\partial \mathbf{z}}(\mathbf{x}, \mathbf{z}). \tag{4}$$

Since $\frac{\partial s_\alpha}{\partial s} = 1$, we can substitute Eq. 2 into Eq. 4 to derive that

$$\frac{\partial v}{\partial \mathbf{z}} = \frac{\partial \mathcal{A}_\Phi}{\partial \mathbf{z}}(\mathbf{x}, \mathbf{z}) - \frac{\partial \mathcal{A}_\Phi}{\partial \mathbf{x}} \nabla s(\mathbf{x}, \mathbf{z}) \frac{\partial s}{\partial \mathbf{z}}(\mathbf{x}, \mathbf{z}). \tag{5}$$

2.4 Garment Draping

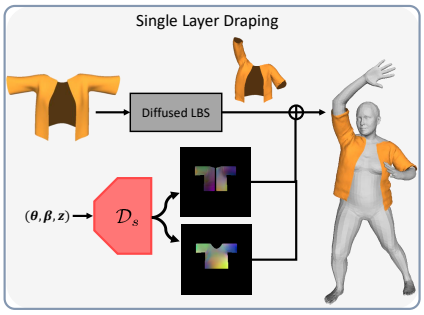

Figure 16: **Single layer draping.** The rest-state garment is first deformed by the diffused LBS to get the initial shape, and then refined by the displacement maps predicted by $\mathcal{D}_s$.

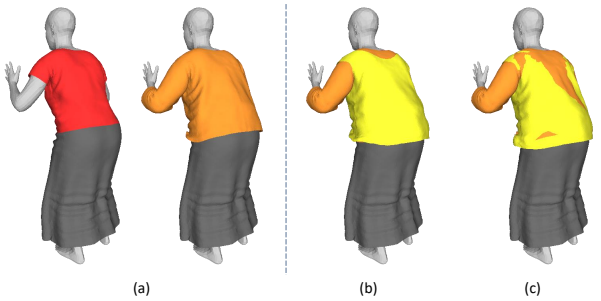

Figure 17: **Layering.** (a) We drape a red and an orange shirts on the body. A third one (yellow) is draped by (b) Algorithm 1 and (c) by naively applying $\mathcal{D}_m$ to it.

Single Layer Draping. In Fig. 16, we illustrate the pipeline for the single layer draping, which relies on the diffused LBS of SMPL [6] to get the initial rough estimate of the garment shape and the displacement map output by $\mathcal{D}_s$ to refine it.

Multi-Layer Draping. Our layering network $\mathcal{D}_m$ can be applied to multiple garments iteratively to resolve collisions between them. More specifically, consider K garments $[G_1, G_2, ..., G_K]$ that are already draped individually by single layer draping network $\mathcal{D}_s$. Their subscripts denote their draping order, with smaller ones being closer to the body. We can obtain their rest state maps $[M_r^1, M_r^2, ..., M_r^K]$ as described in Sec. 3.2 of the main paper, and apply Algorithm 1 for layering them by iterating on garments following their draping order.

Note that we only train $\mathcal{D}_m$ on *one* pair of garments individually draped by $\mathcal{D}_s$. Therefore, it can only resolve the intersections happening at the same layer, which leads to an extra inner loop in Algorithm 1 that moves all the subsequent garments to the same j-th layer. Otherwise, intersections cannot be completely resolved when applying $\mathcal{D}_m$ to two garments lying on the different layers as illustrated in Fig. 17(c).

Algorithm 1: Multi-layer Draping

Require : Function `ForceMap(`a`, `b`)` that computes the force map for a by taking b as the underlying layer; Function `PositionMap(`a`)` that computes the 2D position map of a; Layering network $\mathcal{D}_m$.

Input : An ordered set of garments $[G_1, G_2, ..., G_K]$; Rest state maps for each garemnts $[M_r^1, M_r^2, ..., M_r^K]$.

Output : Layered garments $\{\tilde{G_1}, \tilde{G_2}, ..., \tilde{G_K}\}$ without intersections.

for $i \leftarrow 1$ **to** K **do**
  $\tilde{G_i} \leftarrow G_i$;
  for $j \leftarrow i+1$ **to** K **do**
    $M_f^j \leftarrow$ `ForceMap(`G_j, $\tilde{G_i}$`)` ; `/* Eq. (9) in the main paper */`
    $M_d^j \leftarrow$ `PositionMap(`G_j`)`;
    Update vertex positions of G_j by $\mathcal{D}_m(M_r^j, M_d^j, M_f^j)$;

Figure 18: Segmentation masks obtained from [7]. Jackets are marked in gray, shirts in purple, skirts in dark purple, trousers in red, and body parts in other colors.

2.5 Recovering Multi-Layered Garments from Images

In Fig. 18, we show the segmentation masks used for the optimization in Eq. (11) in the main paper. The optimization is performed from the outer garment to the inner one, i.e., jacket (if detected) $\rightarrow$ shirt $\rightarrow$ trousers (or skirt). More specifically, for the first example shown in Fig. 18, we have the detected segmentation mask $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$ for the jacket, the shirt and the trousers respectively. We first initialize a latent code $\mathbf{z}_1$ for the jacket and perform the following optimization to recover its mesh

$$\mathbf{z}_1^* = \arg\min_{\mathbf{z}_1} L_{\text{IoU}}(R(\mathcal{G}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_1) \oplus \mathcal{M}(\mathbf{B}, \mathbf{\Theta})), \mathbf{S}_1) \ , \tag{6}$$

$$\mathcal{G}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_1) = \mathcal{D}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_1, \mathbf{T}_G(\mathbf{z}_1)) \ , \tag{7}$$

Note that the rendered mask is obtained by setting the colors of the jacket and the body mesh vertices to white and black respectively. Then we fix $\mathbf{z}_1$ and initialize a new latent code $\mathbf{z}_2$ for the shirt and perform

$$\mathbf{z}_2^* = \arg\min_{\mathbf{z}_2} L_{\text{IoU}}(R(\mathcal{G}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_{1:2}) \oplus \mathcal{M}(\mathbf{B}, \mathbf{\Theta})), \mathbf{S}_2) \ , \tag{8}$$

$$\mathcal{G}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_{1:2}) = \mathcal{D}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_1, \mathbf{T}_G(\mathbf{z}_1)) \oplus \mathcal{D}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_2, \mathbf{T}_G(\mathbf{z}_2)), \tag{9}$$

to recover the mesh of the shirt. The optimization for the trousers is similar: fixing $\mathbf{z}_1$ and $\mathbf{z}_2$; initializing the latent code $\mathbf{z}_3$ for the trousers; performing

$$\mathbf{z}_3^* = \arg\min_{\mathbf{z}_3} L_{\text{IoU}}(R(\mathcal{G}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_{1:3}) \oplus \mathcal{M}(\mathbf{B}, \mathbf{\Theta})), \mathbf{S}_3) \ , \tag{10}$$

$$\mathcal{G}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_{1:3}) = \mathcal{D}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_1, \mathbf{T}_G(\mathbf{z}_1)) \oplus \mathcal{D}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_2, \mathbf{T}_G(\mathbf{z}_2)) \oplus \mathcal{D}(\mathbf{B}, \mathbf{\Theta}, \mathbf{z}_3, \mathbf{T}_G(\mathbf{z}_3)). \tag{11}$$

2.6 Loss Terms, Network Architectures and Training

Loss Terms. The Chamfer distance loss $\mathcal{L}_{CHD}$ of Eq. (4) in the main paper is formulated as

$$\mathcal{L}_{CHD} = \sum_{\mathbf{x} \in P} \min_{\mathbf{X} \in \mathcal{S}} ||\mathcal{A}_\Phi(\mathbf{x}, \mathbf{z}) - \mathbf{X}||_2^2 + \sum_{\mathbf{X} \in \mathcal{S}} \min_{\mathbf{x} \in P} ||\mathcal{A}_\Phi(\mathbf{x}, \mathbf{z}), \mathbf{X}||_2^2, \tag{12}$$

Network	$\mathcal{I}_\Theta$	$\mathcal{A}_\Phi$	w	$\mathcal{D}_s$	$\mathcal{D}_m$
Learning Rate	10^{-4}	10^{-4}	10^{-4}	5×10^{-5}	10^{-4}
Batch Size	50	50	6000	30	6
Iterations	70000	70000	2000	20000	30000

Table 4: Training hyperparameters.

Loss	$\mathcal{L}_\mathcal{I}$		$\mathcal{L}_\mathcal{A}$		$\mathcal{L}_{\mathcal{D}_m}$		
	λ_{CE}	λ_{reg}	λ_n	λ_c	λ_g	λ_r	ϵ
Value	0.01	0.001	0.01	1	0.5	1	0.005

Table 5: Training loss hyperparameters for $\mathcal{L}_\mathcal{I}$, $\mathcal{L}_\mathcal{A}$ and $\mathcal{L}_{\mathcal{D}_m}$.

where P is the panel and $\mathcal{S}$ is the ground truth surface mesh.

The normal consistency loss $\mathcal{L}_{normal}$ of Eq. (4) in the main paper is formulated as

$$\mathcal{L}_{normal} = \sum_{\mathbf{x} \in Fc_P} (1 - n_f(\mathcal{A}_\Phi(\mathbf{x}, \mathbf{z})) \cdot \mathbf{n}_{\mathbf{X}^*}) + \sum_{\mathbf{X} \in Fc_\mathcal{S}} (1 - n_f(\mathcal{A}_\Phi(\mathbf{x}^*, \mathbf{z})) \cdot \mathbf{n}_\mathbf{X}), \tag{13}$$

$$\mathbf{X}^* = \underset{\mathbf{X} \in Fc_\mathcal{S}}{\operatorname{argmin}} ||\mathbf{X} - \mathcal{A}_\Phi(\mathbf{x}, \mathbf{z})||_2, \;\; \mathbf{x}^* = \underset{\mathbf{x} \in Fc_P}{\operatorname{argmin}} ||\mathbf{X} - \mathcal{A}_\Phi(\mathbf{x}, \mathbf{z})||_2 \tag{14}$$

where Fc_P and $Fc_\mathcal{S}$ are the face centers of the panel mesh and the surface mesh, and $\mathbf{n}_\mathbf{X}$ represents the normal of $\mathbf{X}$. $n_f(\cdot)$ is the function that computes the normal for the face that $\mathcal{A}_\Phi(\mathbf{x}, \mathbf{z})$ belongs to.

Network Architectures. For each garment category, i.e. shirts, skirts, and trousers, we train one separate set of networks $\{\mathcal{I}_\Theta, \mathcal{A}_\Phi, \mathcal{D}_s\}$. $\mathcal{D}_m$ is shared by all garment categories. Our models are implemented as the following.

- Pattern parameterization network $\mathcal{I}_\Theta$: We use two separate networks $\mathcal{I}_{\Theta_f}$ and $\mathcal{I}_{\Theta_b}$ to learn the pattern parameterization for the front and back panels. Each of them is implemented as an MLP with Softplus activations.
- UV parameterization network $\mathcal{A}_\Phi$: We use two separate networks $\mathcal{A}_{\Phi_f}$ and $\mathcal{A}_{\Phi_b}$ to learn the UV parameterization for the front and back surfaces. Both of them have the same architecture, which is a 7-layer MLP with a skip connection from the input layer to the middle and Softplus activations.
- Latent code $\mathbf{z}$: The dimension is 32.
- Diffuse skinning weight model w: A 9-layer MLP with leaky ReLU activations and an extra Softmax layer at the end to normalize the output.
- Displacement network $\mathcal{D}_s$: A 10-layer MLP with a skip connection from the input layer to the middle and leaky ReLU activations.
- Layering network $\mathcal{D}_m$: A U-Net with 4 convolution blocks and 4 deconvolution blocks.

Training. We use the Adam [8] optimizer for training our networks. The batch sizes, the learning rates and the numbers of iterations for training are summarized in Table. 4. The hyperparameters of the training losses are summarized in Table. 5. $\mathcal{I}_\Theta$, $\mathcal{A}_\Phi$, w and $\mathcal{D}_m$ are trained with a TESLA V100 GPU, while $\mathcal{D}_s$ is trained with 3 GPUs. During the training of $\mathcal{D}_m$, we randomly select two garments as the outer and inner layers, and let the model learn to resolve intersections between them.

3 Extension to Sewing Patterns with More Panels

In our experiments, each garment's sewing pattern consists of two panels, the front and the back. However, our ISP can be extended to patterns with any number of panels. For example, we can train

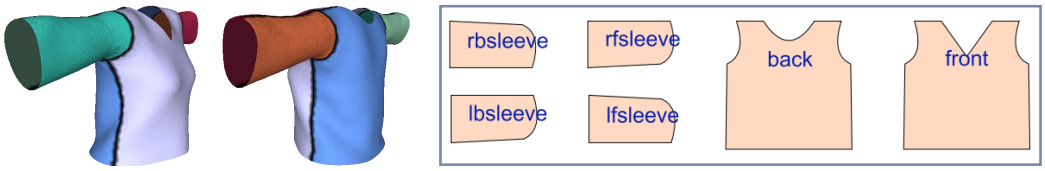

Figure 19: **The 6-panel sewing pattern**. (a) The 3D mesh surface for a shirt. The corresponding surface for each panel is denoted in different colors. (b) The six 2D panels for the front, the back, the right front/back sleeves and the left front/back sleeves.

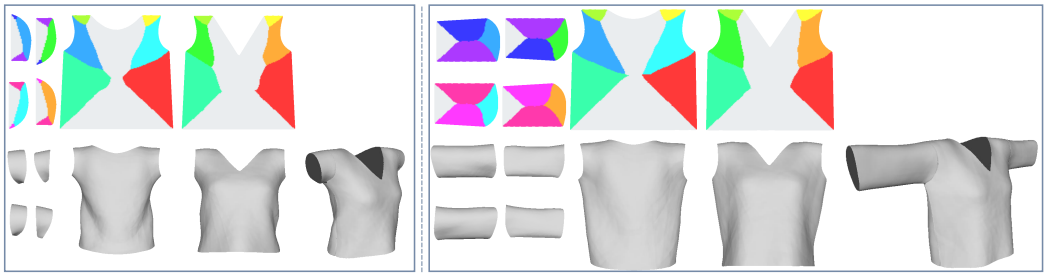

Figure 20: **Reconstruction with 6 panels**. For both the left and right examples, we show the reconstructed panels on the top and the reconstructed surfaces and the sewed meshes at the bottom. Colors on panels denote edge labels predicted by $\mathcal{I}_\Theta$.

$\mathcal{I}_\Theta$ and $\mathcal{A}_\Phi$ on a database of sewing patterns with six panels as shown in Figure 19, using the same training protocol described in the main paper. After training, we can use them to reconstruct the panels and surfaces and produce the sewed mesh as illustrated in Fig. 20. Adding more panels does not result in better reconstructions. For this reason and for the sake of simplicity, we use a model with 2 panels as our default setting.

4 Failure Cases

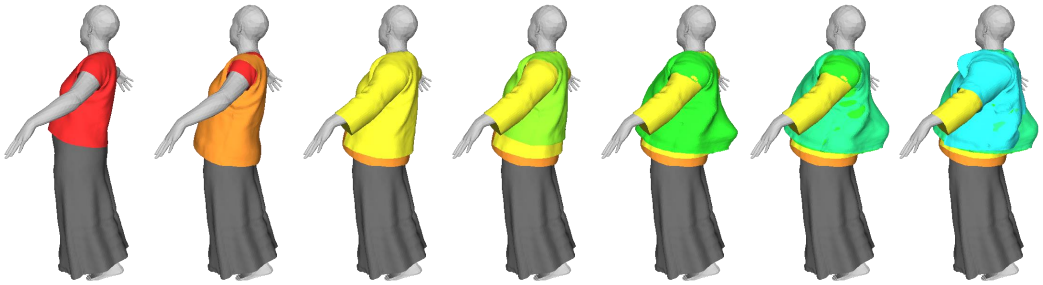

Figure 21: Draping increasingly many shirts: from 1 (left) to 7 (right).

Fig. 21 presents draping results as the number of shirts increases. We observe that the model produces unrealistic deformation when the number of shirts is greater than four. This behavior occurs because our multi-layer draping model $\mathcal{D}_m$ is only trained on garments obtained by single layer draping as described in Section 3.2 of the main paper. In this scenario, the garments are relatively close to the body. When applied to cases with more shirts (typically over four), the model may generate unpredictable results with the shirts moving far away from the body. However, we note that this issue can be resolved by finetuning the model progressively on layered garments. We also consider that wearing more than four shirts is not a common scenario.

References

[1] Luca De Luigi, Ren Li, Benoît Guillard, Mathieu Salzmann, and Pascal Fua. DrapeNet: Generating Garments and Draping them with Self-Supervision. In *Conference on Computer Vision and Pattern Recognition*, 2023.

[2] T. Groueix, M. Fisher, V. Kim, B. Russell, and M. Aubry. Atlasnet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[3] R. Li, B. Guillard, E. Remelli, and P. Fua. DIG: Draping Implicit Garment over the Human Body. In *Asian Conference on Computer Vision*, 2022.

[4] I. Santesteban, M.A. Otaduy, and D. Casas. SNUG: Self-Supervised Neural Dynamic Garments. In *Conference on Computer Vision and Pattern Recognition*, 2022.

[5] E. Remelli, A. Lukoianov, S. Richter, B. Guillard, T. Bagautdinov, P. Baque, and P. Fua. Meshsdf: Differentiable Iso-Surface Extraction. In *Advances in Neural Information Processing Systems*, 2020.

[6] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M.J. Black. SMPL: A Skinned Multi-Person Linear Model. *ACM SIGGRAPH Asia*, 34(6), 2015.

[7] P. Li, Y. Xu, Y. Wei, and Y. Yang. Self-Correction for Human Parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):3260–3271, 2020.

[8] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimisation. In *International Conference on Learning Representations*, 2015.