Supplementary Information for Structure-Preserving 3D Garment Modeling with Neural Sewing Machines

1 Ablations for Reconstruction from the Sewing Pattern

Unified Sewing Pattern Encoding. To verify the indispensability of our unified sewing pattern encoder, in Table 1, we report the results of using different sewing pattern encoding strategies in the task of 3D garment reconstruction from sewing patterns.

We compare sewing pattern encoding strategies at three hierarchical levels: a panel level, a basic panel group level (i.e., our final adoption), and a garment level. For the panel level, we compute embeddings for each panel class. For the basic panel group level, we compute embeddings for each basic panel group. For the garment level, we compute embeddings for each garment category. As we can see, the embeddings of the basic panel group level achieve better performance than the rest two strategies for two metrics (i.e., Chamfer error and P2S error) and achieve comparable performance to the garment level embedding for the other metric (i.e., MGLE error). These comparisons fully demonstrate the rationality and effectiveness of our unified encoder design.

Table 1: Chamfer, P2S errors, and MGLE (cm) of different sewing pattern encoding strategies for 3D garment reconstruction from sewing patterns on the 3D garment dataset.

	Method	t-shirt	jacket	dress	skirt	jumpsuit	pants	All
Chamfer ↓	Panel	1.91	2.61	1.84	1.74	0.72	2.02	1.92
	Basic panel group	1.48	2.17	1.77	1.61	0.66	1.53	**1.65**
	Garment	2.37	3.69	2.17	2.23	0.95	2.05	2.43
P2S ↓	Panel	1.30	2.86	1.29	1.29	0.81	2.99	1.78
	Basic panel group	1.17	2.08	1.27	1.19	0.73	2.13	**1.46**
	Garment	1.67	3.96	1.57	1.44	0.93	2.79	2.17
MGLE ↓	Panel	4.30	5.27	5.37	4.19	3.26	3.07	4.51
	Basic panel group	3.13	4.13	4.26	3.46	2.32	2.65	3.54
	Garment	3.37	3.79	4.01	3.70	1.75	2.59	**3.44**

3D Garment Decoding. To verify the effectiveness of our 3D garment encoder and gain more insight into its ability to model generic garments, in Table 2, we report the results of different 3D garment decoding strategies for our NSM. Specifically, we compared two strategies, i.e., a multi-model strategy and a unified-model strategy (i.e., our final adoption). For the multi-model strategy, we train multiple models to individually/separately perform 3D garment decoding for different garment categories, with each garment category having a specific model for decoding. For the unified-model strategy, we train a single/unified model to universally perform 3D garment decoding for all garment categories. As we can see, the unified-model strategy achieves far better performance than the multi-model strategy for one metric (i.e., 3.54cm vs. 4.36cm for MGLE error) and achieves comparable performance to the multi-model strategy for the other two metrics (i.e., 1.63cm vs. 1.65cm for Chamfer error and 1.43 vs. 1.46cm for P2S error). These comparisons fully demonstrate the rationality and effectiveness of our 3D garment decoder design.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Table 2: Chamfer, P2S errors, and MGLE (cm) of different 3D garment decoding strategies for 3D garment reconstruction from sewing patterns on the 3D garment dataset.

	Method	t-shirt	jacket	dress	skirt	jumpsuit	pants	All
Chamfer ↓	Multi-model	1.54	2.11	1.78	1.49	0.65	1.48	**1.63**
	Unified-model	1.48	2.17	1.77	1.61	0.66	1.53	1.65
P2S ↓	Multi-model	1.27	1.89	1.31	1.16	0.79	1.95	**1.43**
	Unified-model	1.17	2.08	1.27	1.19	0.73	2.13	1.46
MGLE ↓	Multi-model	4.16	4.97	5.20	3.90	3.30	3.33	4.36
	Unified-model	3.13	4.13	4.26	3.46	2.32	2.65	**3.54**

Table 3: Sensitivity analysis of sample points number K in MGLE metric for 3D garment reconstruction from sewing pattern on the 3D garment dataset.

K	2	10	20	40	80	1000
Time (h)	1.38	1.8	2.3	3.6	7.6	10.4
MGLE	3.52	3.53	3.54	3.53	3.52	3.51

2 Sensitivity Analysis of Hyperparameter in MGLE Metric

We conduct sensitivity analysis to the sample points number K in the MGLE metric. We use the task of 3D garment reconstruction from the sewing pattern for validation. The table below shows that the performance is not sensitive to K. Moreover, we can see that K greatly affects the evaluation time. For example, when $K = 1,000$, the evaluation time is already more than 10 hours. Hence, we didn't provide the results of K greater than 1,000 due to the computational cost and unnecessity.

3 Details for 3D Reconstruction from a 2D Single-View Image

Pipeline. The pipeline to implement 3D garment reconstruction from a 2D single-view image is shown in Figure 1. ***First***, given an image, one neural network E will predict its potential embeddings under each basic panel group, and another neural network H will predict which basic panel groups it might contain in terms of a multi-hot vector. ***Second***, the sewing pattern embeddings are obtained by multiplying the predicted basic panel group embeddings by the multi-hot vector. ***Third***, the UV-position maps with masks are predicted from the sewing pattern embedding using our 3D garment decoding module.

Implementation Details. ***First***, we train an NSM with pairs of *{sewing pattern, 3D garment annotation}*. ***Second***, we train a network E with pairs of *{image, sewing pattern embeddings}* using the L_1 loss, where *sewing pattern embeddings* are only available for basic panel groups that appear in the image. ***Third***, we train a network H with pairs of *{image, basic panel group classes}* using the binary cross-entropy loss. We use the architecture of the encoder in [3] as the backbones for networks E and H.

4 3D Garment Readout from UV-Position Maps with Masks

We first convert the points on the UV-position maps into 3D space, as the UV-position maps store the 3D coordinates of the 3D points at their UV coordinates. Then the inner-panel topology is recovered by connecting the 3D adjacent points based on their adjacencies in the UV-position maps, and the inter-panel topology is recovered by utilizing the pre-defined stitch information from the 3D garment dataset, as shown in Figure 2.

Specifically, we first construct the triangulated mesh for each panel. The grid points on the UV-position maps are chosen as the inner vertices if they are inside the panel contour, then we sample the points on the panel contour as the edge vertices. We uniformly sample fixed number of vertices on each contour edge. Then we construct the triangulated mesh from these vertices on 2D plane by automatic triangulation algorithm. The 3D coordinate of each vertex is obtained by bilinear interpolation on the UV-position map.

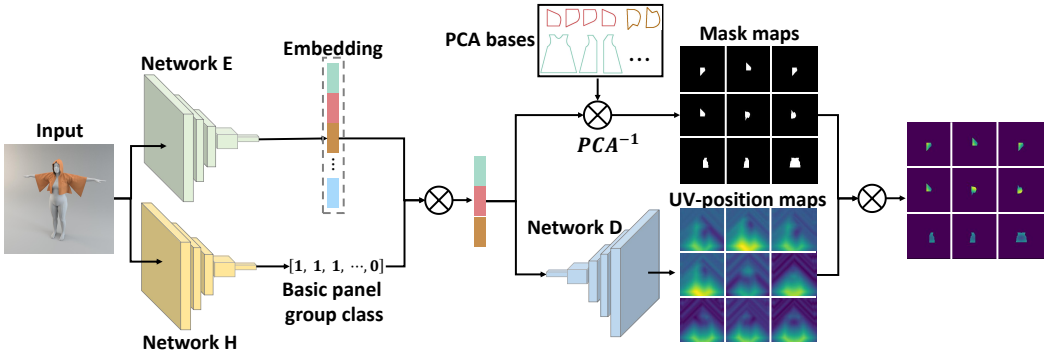

Figure 1: Framework for single-view 3D reconstruction. Given an input image, the network E predicts the embeddings, and the network H predicts the basic panel group classes. Then, the UV-position maps with masks are predicted from the chosen embeddings using our 3D garment decoding module.

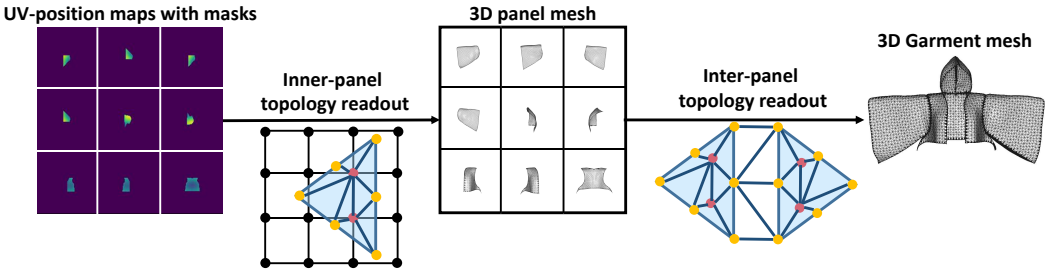

Figure 2: Illustration of a 3D garment readout process. We first convert the points on the UV-position maps into 3D space. Then, the inner-panel topology is recovered by connecting the 3D adjacent points based on their adjacencies in the UV-position maps, and the inter-panel topology is recovered by utilizing the pre-defined stitch information.

Then we construct the inter-panel triangulated mesh. As the stitching information is known, we first determine which two edge from panels are stitched. In general, the panels are stitched uniformly along the edge. As we have uniformly sampled fixed number of vertices on each edge. We also apply the automatic triangulation algorithm to obtain the inter-panel triangulated mesh.

5 3D Garment Draped on Human Body

We present the visualization results of the predicted 3D garments draped on human body, as shown in Figure 3. We adopt simple post processing to prevent the inter-penetration and generate realistic wrinkles by simulating the garment on MAYA for about 1 second. In construct, directly simulating a 3D garment from sewing pattern requires about 30 seconds.

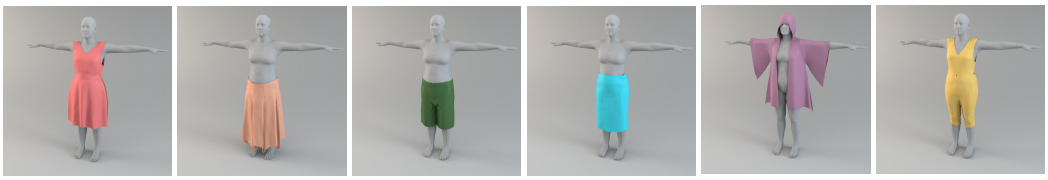

Figure 3: Visualization results of the predicted 3D garments draped on human body.

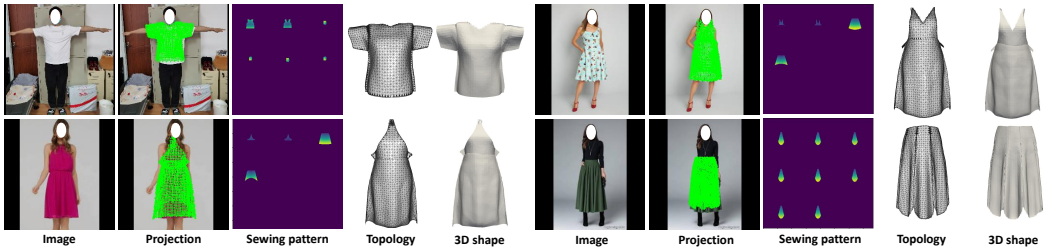

Figure 4: Visualization of results on real images collected from the Internet.

6 Fitting on Real Images

To demonstrate the generalization ability of NSM for real scenes, we used the real-captured in-the-wild images for evaluation. Given an image of a person, we first estimate the camera parameters using [2] and a 2D cloth semantic segmentation using [1], then we fit the trained NSM to the image to obtain the 3D garment. We set the input embedding for the NSM decoder as learnable variables and fixed the NSM decoder parameters. We optimized the projection of the predicted garment to match the cloth segmentation on the image. After fitting the projection of the 3D garment to match cloth segmentation, we obtain the optimized sewing pattern and the 3D garment. The visualization results are shown in Figure 4. Although this is a challenging task (as our NSM is trained only on synthetic data, and there exists a domain gap between synthetic data and real scene images), the promising results in Figure 4 confirm the generalization ability of our method.

7 Video Demos

More **intuitive**, more **enjoyable**, and more **commendable** results are presented in supplementary material in the form of **VIDEO DEMOS**. Please refer to the **VIDEO DEMOS** for more detail and enjoy them.

References

[1] Gong Ke, Gao Yiming, Liang Xiaodan, Shen Xiaohui, Wang Meng, and Lin Liang. Graphonomy: Universal human parsing via graph transfer learning. 2019.

[2] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. Vibe: Video inference for human body pose and shape estimation. 2020.

[3] 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*, pages 234–241. Springer, 2015.