Panelformer: Sewing Pattern Reconstruction from 2D Garment Images

Cheng-Hsiu Chen[1], Jheng-Wei Su[1], Min-Chun Hu[1], Chih-Yuan Yao[2], Hung-Kuo Chu[1,†]
[1]National Tsing Hua University, [2]National Taiwan University of Science and Technology
[†]hkchu@cs.nthu.edu.tw

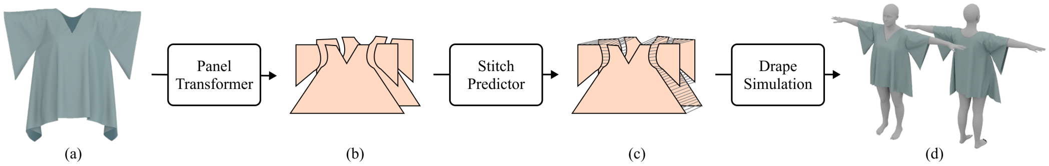


Figure 1. **Overview of the reconstruction process.** We introduce a learning-based approach for reconstructing sewing patterns from garment images. Given (a) and input garment image, we utilize a *panel transformer* to extract (b) the panels of the sewing pattern. Subsequently, we employ a *stitch predictor* to determine (c) the stitching information, resulting in a complete reconstructed sewing pattern. The reconstructed pattern can be further (d) simulated on virtual human bodies through software draping simulation.

Abstract

In this paper, we present a novel approach for reconstructing garment sewing patterns from 2D garment images. Our method addresses the challenge of handling occlusion in 2D images by leveraging the symmetric and correlated nature of garment panels. We introduce a transformer-based deep neural network called Panelformer that learns the parametric space of garment sewing patterns. The network comprises two components: the panel transformer and the stitch predictor. The panel transformer estimates the parametric panel shapes, including the occluded panels, by learning from the visible ones. The stitch predictor determines the stitching information among the predicted panels, enabling the reconstruction of the complete garment. To mitigate the overfitting problem caused by strong panel correlations, we propose two tailor-made data augmentation techniques: panel masking and garment mixing. These techniques generate a wider variety of panel combinations, enhancing the model's robustness and generalization capability. We evaluate the effectiveness of Panelformer using a synthetic dataset with diverse garment types. The experimental results demonstrate that our method outperforms competing baselines and achieves comparable performance to NeuralTailor, which operates on 3D point cloud data. This validates the efficacy of our approach in the context of garment sewing pattern reconstruction. By utilizing 2D images as input, our method expands the potential applications of garment modeling and offers easy accessibility to end users. Our code is available online[1].

1. Introduction

The computer graphics and computer vision community has long been captivated by research about garments modeling due to its wide range of applications, including virtual try-on, avatar generation, and garment design. Several researchers have dedicated their efforts to reconstructing 3D garment models from input garment images using various techniques. Some of these approaches involve fitting predefined garment template models [2,3,6,13,20,31,33] or optimizing sets of parametric sewing patterns [11, 37]. While these methods have demonstrated impressive reconstruction quality, they are primarily limited to pre-defined garment types. To improve their generalizability, researchers have explored the use of robust implicit function representations to faithfully reconstruct garments with varying styles and types [5, 21, 40, 41]. However, the reconstructed garment models often exhibit physical deformations derived from the input images, which may not be desirable when draping garments on different body shapes or poses.

To address the aforementioned issues, NeuralTailor [17] presents a novel deep-learning framework to recover a structured representation of garment sewing patterns from a 3D point cloud. This sewing pattern representation consists

[1] https://ericsujw.github.io/Panelformer/

of a collection of 2D panel shapes, their relative positions in relation to a reference body, and information on how these panels are stitched together to form the final garment. By adopting such a sewing pattern representation, which mimics the fabrication process of real-world garments, NeuralTailor effectively disentangles the overall garment shape from physical deformations, allows for describing a wide range of garment types, and facilitates the sharing of knowledge across different garment types during the training.

Inspired by NeuralTailor [17], this paper introduces a novel approach that focuses on reconstructing garment sewing patterns from 2D garment images instead of 3D point clouds. By utilizing 2D images as input, this approach expands the potential applications as 2D images are easily accessible to end users. However, working with 2D images presents a significant challenge in handling occlusion since nearly half of the garment (specifically, the back-side) is not visible in the projected 2D space.

To tackle this challenge, we leverage the key observation that garment sewing patterns often consist of symmetric panels and exhibit strong correlations between panel shapes. For example, the structure of the left-front sleeve panel is similar to that of the left-back sleeve panel. Based on this insight, we introduce a transformer-based deep neural network called Panelformer to learn the parametric space of garment sewing patterns from input garment images. As shown in Figure 1, our network consists of two main components. The first component, the *panel transformer*, is responsible for estimating the parametric panel shapes of the sewing pattern. Through attention mechanisms, the network can infer the shape of occluded panels (e.g., back sleeves) by learning from the visible ones (e.g., front sleeves). Subsequently, we employ a *stitch predictor* that determines the stitching information among the predicted panels. This component helps establish the connections between different panels, enabling the reconstruction of the complete garment.

Furthermore, we identified a potential overfitting problem during training, stemming from the strong correlation among panels in sewing patterns. For example, suppose the training dataset only includes one type of garment with a hood. In that case, the model might consistently produce the corresponding sewing pattern whenever it encounters a hood, failing to generalize to other garment types. To address this issue, we introduce two tailor-made data augmentation techniques for garment shapes: *panel masking* and *garment mixing*. These techniques enable generating a wider variety of panel combinations, enhancing the model's robustness, and reducing the risk of overfitting.

We perform a comprehensive evaluation to assess the effectiveness of Panelformer using a synthetic dataset comprising a diverse range of garment types. The experimental results demonstrate that our method outperforms competing baselines and is comparable to NeuralTailor [17], which operates on 3D point cloud data. This demonstrates the efficacy of our method in the context of garment sewing pattern reconstruction.

In summary, we make the following contributions:

- We introduce a novel end-to-end transformer architecture for the reconstruction of garment sewing patterns from 2D garment images. To the best of our knowledge, our work represents one of the pioneering approaches in estimating structured sewing patterns solely from 2D images.
- We propose two tailor-made data augmentation techniques to generate a wider variety of panel combinations, thereby enhancing the model's robustness and generalization capability.
- We achieve state-of-the-art performance compared to competing baselines and comparable performance to methods that operate on 3D point cloud data.

2. Related work

Garment modeling from 2D images. Template-based methods are often used to estimate the geometry of an input garment image. Several approaches [11, 37] utilize parametric sewing patterns as templates and optimize the parameters to achieve a simulated result similar to the input garment. With advancements in deep learning techniques within the fields of computer vision and computer graphics, recent works [6, 33] are capable of directly estimating the parameters of 3D garment templates from the given input, thereby accelerating the inference process. Nevertheless, the varying parameter requirements across templates impose limitations on the generalization of these models.

SMPL [19] has emerged as a popular solution to address these issues, owing to its learning-friendly characteristics. SMPL is a 3D human body model parameterized by pose and shape parameters learned from an extensive dataset of human body scans. Several studies [2, 3, 13, 20, 31] have designed garment templates based on the deformation of a submesh of the SMPL body. However, the fixed topology nature of SMPL limits the ability to represent various garments. Recent advancements [5, 21, 40, 41] explore the combination of implicit functions with explicit mesh representation to reduce the necessity of creating new templates.

Another line of research focuses on modeling the dynamics of the garments, which is useful to visualize the quality of reconstructed garments and enable new applications such as virtual try-on. Several works model the task as a function of pose and shape parameters, and directly predict vertex deformations on fixed topology garment meshes [24, 27–29, 34] Alternatively, some works pre-

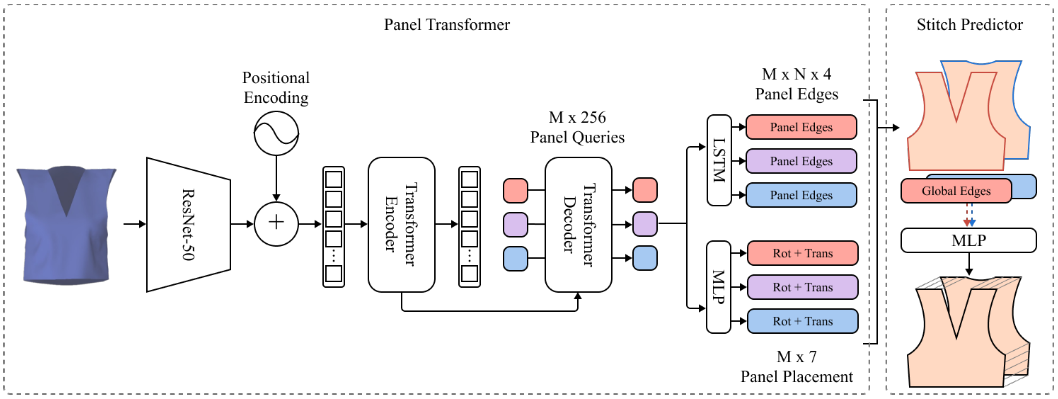


Figure 2. **Network architecture.** The architecture consists of two main components: the panel transformer and the stitch predictor. The input image I is first feed into our panel transformer to reconstruct the shapes and placement of panels. We later use an MLP classifier to predict the stitching information to create a complete sewing pattern.

dict deformations within the 2D image space through carefully designed texture maps [14, 35], harnessing the robustness of CNN architectures. Sewing patterns inherently align with traditional simulators and subsequently generate high quality results without being constrained to pre-defined garment types and topologies.

Template-free rest shape modeling from 3D garments. One commonly employed technique for estimating the rest shape of a 3D garment is surface flattening, wherein the input mesh is cut into developable surfaces and unfolded onto 2D planes [1, 10, 18, 25]. However, these methods require the input mesh to be clean and complete to achieve good results. On the other hand, recent works [17, 26] utilize a novel sewing pattern representation for garment rest shape that exhibits generalizability across various garment types. Their deep learning architecture predicts sewing patterns based on point cloud inputs.

Nevertheless, these existing methods rely on 3D data as input, which is not suitable for our scenario. We propose a novel approach that directly estimates sewing patterns from 2D images. Our experimental results show that our proposed method achieves superior performance compared to the naive approach of first project 2D garment images to 3D and subsequently applying the aforementioned techniques.

Transformer for vision tasks. Transformers have demonstrated remarkable performance in various vision tasks, such as image classification [8] and object detection [4], despite their original design for sequence-to-sequence machine translation tasks [32]. DETR [4] introduced a novel formulation for object detection by utilizing a set of learned object queries to predict bounding boxes. This architecture has been further extended to predict lines [36], planes [30], and polygons [38].

In a similar vein, we adopt a similar formulation for estimating garment patterns. However, unlike RoomFormer [38], which directly regresses the vertices of the polygon, we leverage an LSTM instead of a feed-forward network to generate an indefinite number of vertices. This design choice reduces the overall complexity while still maintaining strong performance.

3. Method

3.1. Data preprocessing

In the data preprocessing stage, we begin with the garment mesh obtained from Korosteleva and Lee [16, 22]. Each garment mesh is initially centered and scaled to fit within the range of $[-0.5, 0.5]$. Next, we render the front view of the normalized garment mesh by positioning the camera at $[0, 0, 1.6]$ and pointing it in the direction of $[0, 0, -1]$ using orthogonal projection. This process yields the garment image $I \in \mathbb{R}^{3\times H\times W}$.

3.2. Sewing pattern representation

The sewing pattern $S = \{P, G\}$ is represented as a set of individual panels P and a set of stitches G that connect the panels together to form a complete garment.

Panel representation. Our model outputs a set of M panels $\{P\}_{i=1}^{M}$. Each panel $P_i = \{E_i, T_i, R_i\}$ consists of edges E_i that represent the outline of panel shape and $\{T_i, R_i\}$ that represent the 3D placement of the 2D panels. And each panel is classified into a corresponding class based on their location and semantic meaning (e.g., front left sleeve panel, top front panel). This allows us to represent a diverse range of garment designs based on different numbers and shapes of panels. The shape of a 2D panel P is denoted as $E_i = (e_i^j, c_i^j)$, a sequence of N consecutive edges, with each edge being either a straight line or a quadratic Bezier spline, represented as the edge vector e_i^j with the edge curvature c_i^j. We denote the edge vector as

$e_i^j = (e_x, e_y) \in [-1, 1]$ in the normalized device coordinates (NDC) space. The sequence of edges E_i form a closed outline of 2D shape by connecting the starting point of every edge to the end point of the previous edge in sequential order. The starting point of the first edge connects to the origin. This way, we obtain a closed outline of the panel shape. Panels that consist of only zero length edges are ignored. To represent curves, we define the edge curvature $c_i^j = (c_x, c_y) \in [-1, 1]$ as the supplement to the edge vector e_i^j. Edge curvature c_i^j indicates the control point of the quadratic Bezier spline defined within the local edge space. The origin of this local edge space is the starting point of the edge vector e_i^j and (1, 0) is the end point of the edge vector e_i^j. We represent the 3D placement of the panel P as the rotation $R_i = (q_x, q_y, q_z, q_w)$ in quaternion form and the translation $T_i = (t_x, t_y, t_z)$ of the panel in the NDC space.

Stitch representation. Stitches $\{G\}$ are defined as a set of 1-to-1 connections between edge of one panel and an edge of another panel within the same garment. In this work, the prediction of stitches is modeled as a binary classification problem of whether there being a connection between the 2 edges. We denote $p_{(i,m)}^{(j,n)}$ as the probability of the existence of stitch between the j^{th} edge of the i^{th} panel and the n^{th} edge of the m^{th} panel.

3.3. Network architecture

Figure 2 illustrates the architecture of Panelformer, which comprises two main components: the panel transformer and the stitch predictor. The panel transformer is responsible for predicting the shape of panels $\hat{P}$, and solving the occlusion issue by learning correlations between panel shapes via the attention mechanisms, while the stitch predictor predicts the connectivity between the edges across different panels.

Panel transformer. We employ ResNet-50 [12] as our feature extraction backbone. Given the input garment image I, we extract a feature map of resolution 16 × 16. To incorporate positional information into this feature map, we add 2-dimensional sine and cosine positional encodings to each location of the feature map. Based on the original encoder-decoder transformer architecture [32], we utilize the flattened positional encoded features as input to the transformer encoder block. The resulting tokens from the encoder block are then employed for cross-attention within the transformer decoder block. To query all panel classes from the garment image I, we utilize a fixed number of M learnable embeddings as input queries for the transformer decoder. Notably, these queries adhere to a fixed order, signifying the specific panel class associated with each query and guaranteeing deterministic results. The output tokens of the transformer decoder are further processed using two auxiliary decoders: (i) an LSTM-based *panel decoder* that generates the panel edge sequence $\{\hat{E}_i\}_{i=1}^{M}$. The sequential nature of the edge sequence makes the LSTM-based model adept at generating reasonable results. (ii) an MLP-based *placement decoder* that outputs the 3D placement, comprising rotation $\{\hat{R}_i\}_{i=1}^{M}$ and translation $\{\hat{T}_i\}_{i=1}^{M}$, for all M panel classes. The design of the decoders follows the module architectures in NeuralTailor [17].

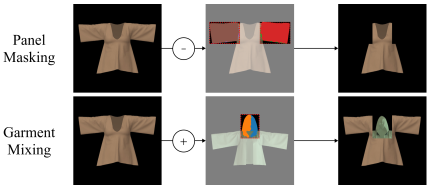


Figure 3. **Data augmentation process.** *Panel masking* removes panels from the garment, and *garment mixing* attaches additional panels to the garment.

Stitch predictor. The edge sequence $\{E_i\}_{i=1}^{M}$ was originally defined in the local panel space. However, panels need to be placed in the same world space in order to perform draping simulation, which is where the stitches come into use. Therefore, we transform the predicted edge sequence $\{\hat{E}_i\}_{i=1}^{M}$ into global edges $\{\hat{V}_i^j = (\hat{u}_i^j, \hat{v}_i^j, \hat{c}_i^j)\}_{j=1}^{N}$ via the predicted 3D placement $\{\hat{T}_i, \hat{R}_i\}_{i=1}^{M}$, where u_i^j and v_i^j are the 3D coordinates of the starting point and ending point of the edge e_i^j in the NDC space, respectively. We then employ a classification MLP as our model and feed all possible global edge pairs $\{(\hat{V}_i^j, \hat{V}_m^n)\}$, where $i \neq m$ and $j \neq n$, into the MLP-based stitch predictor and predict the probability $\{\hat{p}_{(i,m)}^{(j,n)}\}$ of the valid stitches.

3.4. Data augmentation

We introduce two data augmentation techniques inspired by CutOut [7] and Copy-Paste [9], namely *panel masking* and *garment mixing*, as illustrated in Figure 3. Starting from an garment image I, we first acquire the bounding box of each panel leveraging the projected ground truth vertex segmentation of the garment image I. We then assign each panel to a group according to their semantic information. For example, left front sleeve, left back sleeve, right front sleeve, and right back sleeve are assigned to the same group as they all represent sleeves. The groups are denoted as $Q = \{Q_i\}_{i=1}^{L}$, where L is the number of groups. Please refer to our supplementary material for all the details of the groups Q.

Panel masking. We first randomly select one group Q_i within the input image I. For every panel within the chosen group Q_i, the pixels enclosed by their respective bounding box B_i, as well as the corresponding ground truth information, are set to zero.

Garment mixing. We begin by randomly selecting another image I' from the training set. Subsequently, one of the groups Q'_i is randomly chosen from the selected image I'. Following this selection, the pixels contained within the corresponding bounding box B'_i, along with the ground truth information associated with the panels belonging to the chosen group, are copied and integrated into the current input image I.

By applying these data augmentation techniques, we introduce variations and increase the diversity of the training dataset. These augmentation techniques effectively prevent overfitting and improve the robustness of the model.

3.5. Loss functions

Edge loss. The edge loss L_{edge} primarily focuses on ensuring the accurate shape of the predicted panels, formulated as the mean squared error (MSE) between the ground truth panel edge sequence and the corresponding predicted panel edge sequence, as shown below:

$$L_{edge} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}(\hat{E}_{ij} - E_{ij})^2}{M \cdot N \cdot 4}. \tag{1}$$

Loop loss. The loop loss L_{loop} is defined as the L_2 norm of the distance between starting point and the ending point of the predicted panel edge sequence, as shown below:

$$L_{loop} = \frac{1}{M}\sum_{i=1}^{M}(\sum_{j=1}^{N}\hat{e}_i^j)^2, \tag{2}$$

by incorporating the loop loss, we encourage the predicted panel edges to form closed loops, mimicking the characteristic closed structure of panels.

Placement loss. The placement loss consist of L_{rot} and L_{trans}, where both of them defined as the MSE loss between the ground truth and the predicted rotation and translation, as shown below:

$$L_{rot} = \frac{\sum_{i=1}^{M}(\hat{R}_i - R_i)^2}{M \cdot 4}, \tag{3}$$

$$L_{trans} = \frac{\sum_{i=1}^{M}(\hat{T}_i - T_i)^2}{M \cdot 3}. \tag{4}$$

The overall loss function L_{total} for training the shape and placement model is defined as follows:

$$L_{total} = \lambda_1 L_{edge} + \lambda_2 L_{loop} + \lambda_3 L_{rot} + \lambda_4 L_{trans}, \tag{5}$$

where $\lambda_{\{1,\ldots,4\}}$ are the hyperparameters for weighting the loss functions.

Stitch loss. The stitch loss L_{stitch} calculates the binary cross entropy loss between predicted probability $\hat{p}_{(i,m)}^{(j,n)}$ and corresponding ground truth $p_{(i,m)}^{(j,n)}$, as shown below:

$$\begin{aligned} L_{stitch} =& \frac{1}{M^2 \cdot N^2}\sum p_{(i,m)}^{(j,n)} \cdot \log\sigma(\hat{p}_{(i,m)}^{(j,n)}) \\ &+ (1 - p_{(i,m)}^{(j,n)}) \cdot \log(1 - \sigma(\hat{p}_{(i,m)}^{(j,n)})), \end{aligned} \tag{6}$$

where every possible stitch between all edges of all different panels are computed.

4. Experiments

4.1. Experimental settings

Dataset. For both training and evaluation purposes, we utilize the Dataset of 3D garments with sewing patterns [22] obtained from [16]. This dataset comprises a total of 23,500 samples, covering 19 distinct base garment types. Each sample within the dataset contains a garment mesh and the corresponding sewing pattern. As described in Section 3.1, we then render the 2D garment image, which is the front view of the garment mesh for each sample, and use them as input data for all experiments. The samples in the testing set are further divided into two categories: ”unseen types” and ”seen types.” The ”unseen types” category includes samples from seven garment types that do not exist in the training set. On the contrary, the category ”seen types” comprises samples from the remaining 12 garment types that are present in the training set. This division allows us to evaluate the generalizability of our model on both familiar and unfamiliar garment types. We adopt the same filtering process to NeuralTailor [17], where samples exhibiting similar draping results but different sewing pattern arrangements are filtered, resulting in training, validation, and testing sets with 9678, 1200, and 1765 samples, respectively.

Baselines. We conduct a comparative analysis between our method and a naive baseline approach. The baseline approach involves a two-step process: first, predicting 3D point clouds from 2D images, and then using NeuralTailor [17] to reconstruct sewing patterns based on the predicted 3D point clouds (the code of Personaltailor [26] is not available.) For the 3D point cloud prediction, we employ the AnchorUDF model [39], which has been fine-tuned on the test set. Regarding the prediction of sewing patterns, we fine-tune the NeuralTailor architecture [17] using our proposed data augmentation techniques.

Implementation details. Our model is implemented with PyTorch [23]. Training and evaluation processes are performed on a single NVIDIA GeForce RTX 3090 GPU with 24GB VRAM. The input images are resized to a resolution of 512 × 512 pixels and normalized using mean values of [0.485, 0.456, 0.406] and standard deviation values of [0.229, 0.224, 0.225]. The number of panel classes M is set to 23, the maximum number of edges per panel N is set to 14, and the maximum number of groups L is set to 6. The

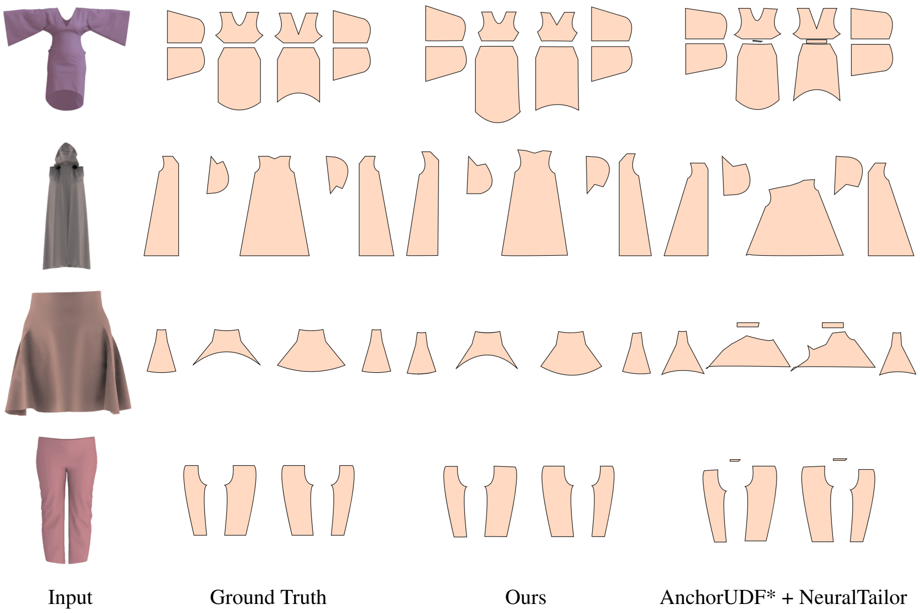


Figure 4. **Qualitative comparison on panel reconstruction.** The top 2 rows show samples from the unseen types testing set and the bottom 2 rows show samples from the seen types testing set. We do not show results from AnchorUDF + NeuralTailor here as the setting hardly produces meaningful shapes.

hidden dimension of the panel transformer is set to 256. We emperically set $\lambda_1 = 1$, $\lambda_2 = 1$, $\lambda_3 = 1$ and $\lambda_4 = 1$ in Equation 5. We use the Adam optimizer [15] with b1=0.9 and b2=0.99. We train the panel transformer for 350 epochs and the stitch predictor for 350 epochs with batch size 32 and 30, respectively. In general, it takes 25 hours to train our Panelformer. We divide our panel transformer training process into two stages, which involve a pre-training step without translation loss L_{trans} followed by a fine-tuning step that incorporates translation loss L_{trans}. In the pre-training step, we set the learning rate to $1e-5$ for the ResNet-50 module [12], while a learning rate of $1e-4$ is used for the remaining modules. The ResNet-50 module is initialized with the pre-trained weights provided by the torchvision library. During the fine-tuning step, we multiply all the learning rates by 0.1 and add translation loss L_{trans} to fine-tune. For the stitch prediction model, we use one-cyclic scheduling with the maximum learning rate set to 0.002. As for the training data, we take two kinds of edge pairs: (i) ground truth edge pairs, and (ii) predicted edge pairs derived from panel transformer.

Metrics. We use the metrics introduced in NeuralTailor [17] to assess the quality of our results. To evaluate the overall quality of the sewing pattern, we compute the accuracy of the number of panels predicted for each pattern (#P) and the number of edges predicted for each panel (#E). To evaluate the reconstructed quality of the shapes of predicted panels, we calculate the L2 norm between the predicted vertices and the ground truth vertices, along with the curvature coordinates (L2-P). We also estimate the L2 norm between the predicted translations (L2-T) and the predicted rotations (L2-R) and their ground truth values. For stitches, we report the precision (Prec.) and recall (Rec.) of the prediction.

4.2. Sewing pattern reconstruction performance

Quantitative comparison on panels. As shown in Table 1, our Panelformer outperforms all the baselines across most of the metrics, except for the rotation error in terms of seen garment types. Compared to AnchorUDF + NeuralTailor, our model improves #P by 89.3% and #E by 95.2%. We observe that NeuralTailor [17] is highly sensitive to the global translation of the input point cloud. It is unable to adapt to the normalized point cloud produced by AnchorUDF, which is fitted within the NDC space. To address this issue, we incorporate the ground-truth global translation into the predicted point cloud from AnchorUDF. We then feed these adjusted point clouds into NeuralTailor. The setting is denoted as AnchorUDF* + NeuralTailor.

Compared to AnchorUDF* + NeuralTailor, our model achieves notable improvements in #P (by 38.8%), L2-P (by 2.3), and #E (by 1.6%), while maintaining a comparable rotation error (L2-R). These results indicate that NeuralTailor is highly sensitive to the imperfect and noisy predicted point clouds obtained from AnchorUDF. In contrast, our Panelformer is an end-to-end model that eliminates the need to predict a 3D point cloud as an intermediate step before

Table 1. **Quantitative comparisons on predicted panels from different baselines.** The best results are in bold font.

Method	Seen Types					Unseen Types				
	L2-P ↓	#P (%) ↑	#E (%) ↑	L2-R ↓	L2-T ↓	L2-P ↓	#P (%) ↑	#E (%) ↑	L2-R ↓	L2-T ↓
AnchorUDF + NeuralTailor	24.4	10.5	4.5	0.34	14.59	25.2	2.6	4.4	0.36	16.48
AnchorUDF* + NeuralTailor	4.2	61.0	98.1	**0.00**	2.49	**5.2**	85.6	90.1	**0.00**	**3.59**
Ours	**1.9**	**99.8**	**99.7**	0.01	**1.78**	5.4	**97.3**	**92.7**	0.01	7.24

transforming it into a sewing pattern.

Table 2. **Quantitative comparisons on predicted stitches from different baselines.** The results from AnchorUDF + NeuralTailor are omitted as the setting hardly produces meaningful shapes.

	Seen Types		Unseen Types	
	Prec. (%) ↑	Rec. (%) ↑	Prec. (%) ↑	Rec. (%) ↑
AnchorUDF* + NeuralTailor on GT	76.0	71.5	68.9	61.5
AnchorUDF* + NeuralTailor on Preds.	70.6	91.1	70.7	**88.0**
Ours on GT	**99.2**	99.6	78.9	79.5
Ours on Preds.	98.9	**99.9**	**83.3**	81.1

Table 3. **Ablation study on various data augmentation.** We compare the results without the fine-tuning with L_{trans}.

	Unseen Types			
	L2-P ↓	#P (%) ↑	#E (%) ↑	L2-R ↓
Ours w/o augmentation	13.5	14.1	60.7	0.12
Ours w/ standard masking	10.2	10.0	80.7	0.13
Ours w/ panel masking	8.6	73.4	84.2	0.03
Ours w/ garment mixing	8.5	65.8	83.9	0.04
Ours	**5.4**	**97.3**	**92.7**	**0.01**

Table 4. **Ablation study on perdicted panels on positional encoding (PE).** We compare the results without L_{trans} fine-tuning.

	Seen Types				Unseen Types			
	L2-P ↓	#P (%) ↑	#E (%) ↑	L2-R ↓	L2-P ↓	#P (%) ↑	#E (%) ↑	L2-R ↓
Ours w/o PE	3.2	99.4	99.6	**0.01**	6.1	79.2	91.2	0.03
Ours w/ PE	**1.9**	**99.8**	**99.7**	**0.01**	**5.4**	**97.3**	**92.7**	**0.01**

As for unseen garment types, our Panelformer surpasses all the baselines in #P and #E, while maintaining comparable performance in other metrics. Compared to AnchorUDF* + NeuralTailor, our model still achieves significant improvements in #P (by 11.7%) and #E (by 2.6%), despite the fact that we favor the baselines by providing ground-truth translations and fine-tuning AnchorUDF on the test set, which includes both seen and unseen types, rather than on the training set, which contains only the seen types from the same dataset. Based on our experiments, the overall quality of the reconstructed sewing patterns largely relies on the accurate prediction of the number of panels (#P) and the number of edges (#E).

Quantitative comparison on stitch prediction. As shown in Table 2, our Panelformer demonstrates superior performance in terms of both precision and recall. This can be attributed to the high quality of our shape predictions, which serve as a foundation for accurate stitch inference. Additionally, we observe that the methods trained on predicted edge pairs perform better on unseen types. We believe it is because imperfect data can enhance the model's robustness.

Qualitative comparison on panel reconstruction. The qualitative comparisons, as depicted in Figure 4, further support the reported metrics. Our Panelformer produces results with fewer redundant panels and more accurate panel shapes compared to AnchorUDF* + NeuralTailor. In general, our model follows an end-to-end approach, eliminating the need for predicting intermediate 3D point clouds, which contributes to greater result stability. Additionally, our data augmentation techniques, such as panel masking and garment mixing, enhance the robustness of our model, enabling reasonable panel reconstruction even for unseen types. Please refer to our supplemental material for more quantitative and qualitative results.

4.3. Ablation study

Effects of different data augmentation. In this experiment, we aim to assess the effectiveness of each augmentation technique used in our model. We compare the setting with all augmentations against different variants to evaluate their impact. The variants include: (i) **Ours w/o augmentation** represents our model without any augmentation techniques applied; (ii) **Ours w/ standard masking** applies random masking of images without the removal of corresponding ground-truth panel; (iii) **Ours w/ panel masking** includes the augmentation technique of *panel masking* only; and (iv) **Ours w/ garment mixing** incorporates the augmentation technique of *garment mixing* only.

The results of this ablation study, as shown in Table 3, show that the implementation of standard masking does not yield noticeable reductions in overfitting. This outcome aligns with expectations, as standard masking merely enhances the visual diversity of the input images without introducing any significant variation in the corresponding ground-truth sewing patterns. On the contrary, both *panel masking* and *garment mixing* significantly increase performance. Both methods contribute to increasing the variety of data and effectively prevent the model from overfitting to the training set. Moreover, *panel masking* outperforms

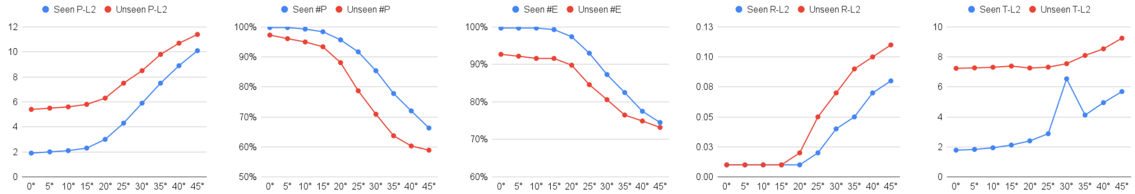
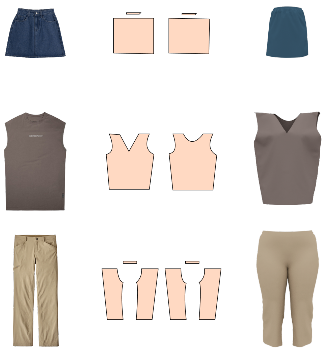


Figure 5. **Performance of each shape reconstruction metric on different viewing angle.** The performance is the best when there is no perturbation and shows noticeable decline beyond 20 degrees.

garment mixing marginally, probably due to the slight difference from the original images before *garment mixing*. Combining the utilization of *panel masking* and *garment mixing* yields the most optimal performance among all experimental settings.

Effects of positional encoding. We compare our original model with a modified version where positional encoding is excluded from the input. The results are presented in Table 4: The omission of positional encoding degrades the performance of L2-P on seen types (by 1.3) and fails to generalize to unseen types, resulting in significant declines on L2-P (by 0.7), #P (by 18.1%), and #E (by 1.5%).

Effects of different viewing angle. In order to assess the robustness of our model against different viewing angles, we generate garment images at different angles ranging from 0 to 45 degrees, with increments of 5 degrees. The predicted results, as depicted in Figure 5, show a slightly drop before 20 degrees. This suggests that our model is capable of tolerating minor perturbations in viewing angles.

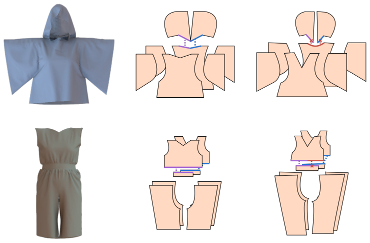


Input Ours Simulation [16]

Figure 6. **Qualitative results on real-world data.** We reconstruct sewing patterns and perform drape simulation for real images.

4.4. Qualitative results on real-world data

We collect real world garment images from the internet and reconstruct their sewing pattern using our Panelformer. We then drape the produced sewing pattern over a human body using software simulation [16]. Our model demonstrate the ability to make accurate estimations regarding the overall structure of the garments. However, it should be noted that finer details of the garments, such as pockets and buttons, were not fully reconstructed in the process. The recovery of these intricate details through techniques like texturing or expanding the range of panel classes remains a crucial task for future endeavors in this field.



Input Ground Truth Ours

Figure 7. **Limitations.** Edge pairs that are colored with the same color indicate that they can be connected by a stitch. The red edges failed to connect to a reasonable edge.

5. Conclusion

We propose Panelformer, an end-to-end model that focuses on reconstructing sewing patterns from 2D garment images. To enhance the model's performance, we introduce augmentation techniques aimed at improving the robustness of our model. The experimental results show the exceptional performance of our model.

Limitation and future work. Our model occasionally predicts unreasonable edge combinations between panels as shown in Figure 7. We plan to develop a model that can jointly predict both the shape and stitch of sewing patterns. Another limitation is that our model can only take one front view image as input. In the future we will move on to utilizing multi-view images as input. Also, the hyper-parameters in Equation 5 have not been thoroughly tuned. We intend to optimize these parameters to improve model performance in future.

References

[1] Seungbae Bang, Maria Korosteleva, and Sung-Hee Lee. Estimating garment patterns from static scan data. *Computer Graphics Forum*, 40(6):273–287, 2021. 3

[2] Hugo Bertiche, Meysam Madadi, and Sergio Escalera. Cloth3d: Clothed 3d humans. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 344–359, Cham, 2020. Springer International Publishing. 1, 2

[3] Bharat Lal Bhatnagar, Garvita Tiwari, Christian Theobalt, and Gerard Pons-Moll. Multi-garment net: Learning to dress 3d people from images. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct 2019. 1, 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing. 3

[5] Enric Corona, Albert Pumarola, Guillem Alenyà, Gerard Pons-Moll, and Francesc Moreno-Noguer. Smplicit: Topology-aware generative model for clothed people. In *CVPR*, 2021. 1, 2

[6] R. Danźřek, E. Dibra, C. Öztireli, R. Ziegler, and M. Gross. Deepgarment: 3d garment shape estimation from a single image. *Comput. Graph. Forum*, 36(2):269–280, may 2017. 1, 2

[7] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 4

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 3

[9] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2918–2928, June 2021. 4

[10] Chihiro Goto and Nobuyuki Umetani. Data-driven Garment Pattern Estimation from 3D Geometries. In Holger Theisel and Michael Wimmer, editors, *Eurographics 2021 - Short Papers*. The Eurographics Association, 2021. 3

[11] Nils Hasler, Bodo Rosenhahn, and Hans-Peter Seidel. Reverse engineering garments. In André Gagalowicz and Wilfried Philips, editors, *Computer Vision/Computer Graphics Collaboration Techniques*, pages 200–211, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. 1, 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 4, 6

[13] Boyi Jiang, Juyong Zhang, Yang Hong, Jinhao Luo, Ligang Liu, and Hujun Bao. Bcnet: Learning body and cloth shape from a single image. In *European Conference on Computer Vision*. Springer, 2020. 1, 2

[14] N. Jin, Y. Zhu, Z. Geng, and R. Fedkiw. A pixel-based framework for data-driven clothing. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '20, Goslar, DEU, 2020. Eurographics Association. 3

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[16] Maria Korosteleva and Sung-Hee Lee. Generating datasets of 3d garments with sewing patterns. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. 3, 5, 8

[17] Maria Korosteleva and Sung-Hee Lee. Neuraltailor: Reconstructing sewing pattern structures from 3d point clouds of garments. *ACM Trans. Graph.*, 41(4), 2022. 1, 2, 3, 4, 5, 6

[18] Kaixuan Liu, Xianyi Zeng, Pascal Bruniaux, Xuyuan Tao, Xiaofeng Yao, Victoria Li, and Jianping Wang. 3d interactive garment pattern-making technology. *Comput. Aided Des.*, 104(C):113–124, nov 2018. 3

[19] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 2

[20] 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 *Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1, 2

[21] 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, October 2021. 1, 2

[22] Korosteleva Maria and Lee Sung-Hee. Dataset of 3d garments with sewing patterns, June 2021. 3, 5

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 5

[24] Chaitanya Patel, Zhouyingcheng Liao, and Gerard Pons-Moll. Tailornet: Predicting clothing in 3d as a function of human pose, shape and garment style. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020. 2

[25] Nico Pietroni, Corentin Dumery, Raphael Falque, Mark Liu, Teresa Vidal-Calleja, and Olga Sorkine-Hornung. Computational pattern making from 3d garment models. *ACM Trans. Graph.*, 41(4), jul 2022. 3

[26] Anran Qi, Sauradip Nag, Xiatian Zhu, and Ariel Shamir. Personaltailor: Personalizing 2d pattern design from 3d garment point clouds, 2023. 3, 5

[27] Igor Santesteban, Miguel A. Otaduy, and Dan Casas. Learning-Based Animation of Clothing for Virtual Try-On. *Computer Graphics Forum (Proc. Eurographics)*, 2019. 2

[28] Igor Santesteban, Miguel A Otaduy, and Dan Casas. SNUG: Self-Supervised Neural Dynamic Garments. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[29] Igor Santesteban, Nils Thuerey, Miguel A Otaduy, and Dan Casas. Self-Supervised Collision Handling via Generative 3D Garment Models for Virtual Try-On. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[30] Bin Tan, Nan Xue, Song Bai, Tianfu Wu, and Gui-Song Xia. Planetr: Structure-guided transformers for 3d plane recovery. In *International Conference on Computer Vision*, 2021. 3

[31] Garvita Tiwari, Bharat Lal Bhatnagar, Tony Tung, and Gerard Pons-Moll. Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. In *European Conference on Computer Vision (ECCV)*. Springer, August 2020. 1, 2

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. 3, 4

[33] Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popovic, and Niloy J. Mitra. Learning a shared shape space for multimodal garment design. *ACM Trans. Graph.*, 37(6):1:1–1:14, 2018. 1, 2

[34] Jane Wu, Zhenglin Geng, Hui Zhou, and Ronald Fedkiw. Skinning a parameterization of three-dimensional space for neural network cloth, 2020. 2

[35] Jane Wu, Yongxu Jin, Zhenglin Geng, Hui Zhou, and Ronald Fedkiw. Recovering geometric information with learned texture perturbations. 4(3), sep 2021. 3

[36] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. Line segment detection using transformers without edges. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4257–4266, June 2021. 3

[37] Shan Yang, Zherong Pan, Tanya Amert, Ke Wang, Licheng Yu, Tamara Berg, and Ming C. Lin. Physics-inspired garment recovery from a single-view image. *ACM Trans. Graph.*, 37(5), nov 2018. 1, 2

[38] Yuanwen Yue, Theodora Kontogianni, Konrad Schindler, and Francis Engelmann. Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[39] Fang Zhao, Wenhao Wang, Shengcai Liao, and Ling Shao. Learning anchored unsigned distance functions with gradient direction alignment for single-view garment reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12674–12683, 2021. 5

[40] Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images, 2020. 1, 2

[41] Heming Zhu, Lingteng Qiu, Yuda Qiu, and Xiaoguang Han. Registering explicit to implicit: Towards high-fidelity garment mesh reconstruction from single images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3845–3854, June 2022. 1, 2