GarmentImage: Raster Encoding of Garment Sewing Patterns with Diverse Topologies

YUKI TATSUKAWA, The University of Tokyo, Japan
ANRAN QI, The University of Tokyo, Japan and Centre Inria d'Université Côte d'Azur, France
I-CHAO SHEN, The University of Tokyo, Japan
TAKEO IGARASHI, The University of Tokyo, Japan

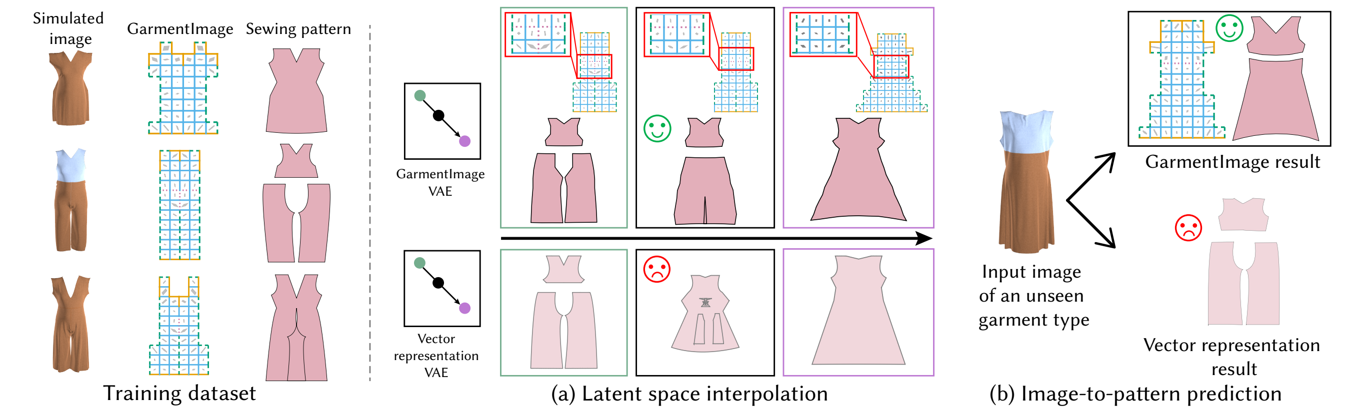

Fig. 1. GarmentImage encodes a garment sewing pattern's geometry, topology and placement as raster data. This leads to a more continuous latent space and improved generalizability to unseen topologies compared to vector-based sewing pattern representation. (a) Interpolation between the two patterns with different topologies (green and purple) in the latent space of the GarmentImage-trained VAE yields a continuous transition and seamless panel merging (top), whereas the vector-based representation-trained VAE generates an invalid pattern (bottom). (b) When given an image of an unseen garment type (*top + skirt*), the GarmentImage-trained model successfully predicts the new pattern (top), whereas the vector-based model defaults to a known pattern (*top + pants*) present in the training data (bottom).

Garment sewing patterns are the design language behind clothing, yet their current vector-based digital representations weren't built with machine learning in mind. Vector-based representation encodes a sewing pattern as a discrete set of panels, each defined as a sequence of lines and curves, stitching information between panels and the placement of each panel around a body. However, this representation causes two major challenges for neural networks: discontinuity in latent space between patterns with different topologies and limited generalization to garments with unseen topologies in the training data. In this work, we introduce GarmentImage, a unified raster-based sewing pattern representation. GarmentImage encodes a garment sewing pattern's geometry, topology and placement into multi-channel regular grids. Machine learning models trained on GarmentImage achieve seamless transitions between patterns with different topologies and show better generalization capabilities compared to models trained on vector-based representation. We demonstrate the effectiveness of GarmentImage across three applications: pattern exploration in latent space, text-based pattern editing, and image-to-pattern prediction. The results show that GarmentImage achieves superior performance on these applications using only simple convolutional networks.

CCS Concepts: • **Computing methodologies** → **Shape modeling**.

Additional Key Words and Phrases: Garment pattern modeling, Garment pattern design

ACM Reference Format:
Yuki Tatsukawa, Anran Qi, I-Chao Shen, and Takeo Igarashi. 2025. GarmentImage: Raster Encoding of Garment Sewing Patterns with Diverse Topologies. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers (SIGGRAPH Conference Papers '25), August 10–14, 2025, Vancouver, BC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3721238.3730632

1 Introduction

A sewing pattern is a pre-designed template to cut fabric pieces for creating a garment or textile item [Aldrich 2015; Armstrong 2014]. It typically consists of a collection of 2D fabric pieces, or *panels*, with annotations specifying connectivity among borders and placements around a body. In the past, a sewing pattern was drawn directly onto fabric or paper using rulers and measuring tapes. Later, in Computer-Aided Design systems, it is presented as a discrete collection of 2D shapes composed of lines and curves. In recent years, many learning-based methods have utilized this vector-based representation for

Authors' Contact Information: Yuki Tatsukawa, The University of Tokyo, Japan, yukitatsu0817@gmail.com; Anran Qi, The University of Tokyo, Japan and Centre Inria d'Université Côte d'Azur, France, anran.qi@inria.fr; I-Chao Shen, The University of Tokyo, Japan, ichaoshen@g.ecc.u-tokyo.ac.jp; Takeo Igarashi, The University of Tokyo, Japan, takeo@acm.org.

various tasks, such as inferring garment sewing patterns from point cloud [Korosteleva and Lee 2022], photograph [Liu et al. 2023], and text [He et al. 2024]. These advancements significantly simplify the process of designing garment sewing patterns.

Despite the widespread use of vector-based representation, we observe two challenges when using it in learning-based methods. **1. Discontinuity in latent space:** In vector-based representation, two patterns with similar overall 2D shapes but different topologies often have different numbers of panels and panel topologies. This causes the latent space created by encoding the vector-based representation to show significant discontinuities between patterns with different topologies (Figure 1(a)). **2. Limited generalization to garments with topologies unseen in the training data:** The above challenge also extends to pattern prediction from a given input, such as image or 3D model of a draped garment. Visually similar inputs can lead to significantly different vector-based pattern representations. This increases the Lipschitz bound of machine learning models trained to predict patterns from inputs, reducing their ability to generalize to patterns with unseen topologies (Figure 1(b)). Additionally, handling variable numbers of panels across multiple garment types requires machine learning models to implicitly map input patterns to a discrete set of panels from a predefined panel pool. This discrete set selection becomes particularly challenging for garments with unseen topologies, further limiting the generalization ability of a trained model.

To address these challenges, we present GarmentImage, a novel raster representation of sewing patterns. GarmentImage integrates the discrete collection of 2D panels, the connectivity among panels, and the placement of panels into multi-channel 2D grids. Each grid cell contains an *inside/outside* flag indicating occupancy, four edges with associated *edge types* embedding stitching information, and a local *deformation matrix* capturing the panel geometry. The placement of a panel around a body is implicitly represented by the location of its associated cells on the 2D grid.

GarmentImage brings several benefits. First, the resulting representation is a 2D matrix of numerical values, similar to a raster image. This enables the use of established raster image generative modeling techniques to model, edit, and optimize sewing patterns. We employ simple convolutional neural networks to encode and decode the GarmentImage, demonstrating that even a straightforward network structure can effectively handle the complex pattern representation and its manipulation. Second, changes in pattern topology are embedded within the grid structure in GarmentImage rather than as discrete panel selection in vector-based representation. This leads to a continuous transition in the latent space between patterns with different topologies, enabling both continuous topology interpolation and prompt-driven pattern optimization directly in the latent space (Figure 1(a)). Third, when inferring the pattern from an input, such as image, we can directly predict the GarmentImage as a whole and then procedurally reconstruct the discrete panel sets, effectively addressing the challenge of discrete panel selection. Additionally, garments with similar visual looks have similar GarmentImage representations regardless of topology. These properties improve the generalization ability of machine learning models trained on GarmentImage, leading to better pattern prediction performance on garments with unseen topologies (Figure 1(b)).

To summarize, the contribution of this work is

- we present GarmentImage, a raster-based sewing pattern representation that integrates geometry, topology and placement information into multi-channel regular grids.
- we showcase its advantages on interpolation and generalization ability over three applications: (1) VAE latent space exploration, (2) text-based pattern editing, and (3) image-to-pattern prediction.

2 Related Work

2.1 Garment Pattern Representation

Garments in our daily lives are composed of various fabric pieces sewn together. To translate this into a digital format, sewing patterns are often represented as a collection of polygons called panels, with stitching details defining how these panels are assembled. The panel is often defined as a closed loop of parametric curves (e.g., B-splines or Bezier curves), defined by vertices and control points.

In computer graphics, the processing of garment panels varies based on the nature of the task. For tasks involving 3D surface data, such as garment simulation [Narain et al. 2012], sewing pattern grading [Brouet et al. 2012; Wang 2018] and sewing pattern adjustment from 3D shape editing [Bartle et al. 2016; Qi and Igarashi 2024] or sketch [Li et al. 2018], panels are often processed as a 2D mesh, which provides computational simplicity and ease of manipulation. For example, to simulate the details of the cloth, such as wrinkles, Narain et al. [2012] presented each panel as a triangulated 2D mesh, and dynamically refined and coarsened triangle meshes to efficiently model the details. On the other hand, for tasks that focus on the panel's shape only, e.g., garment pattern inferences from image [Chen et al. 2024; Jeong et al. 2015; Liu et al. 2023; Su et al. 2020; Yang et al. 2018], sketch [Wang et al. 2018], 3D point cloud [Korosteleva and Lee 2022], a parametrized representation is usually adopted. For example, to predict the sewing pattern of a given image, Yang et al. [2018] parametrized the sewing pattern on several parameters that define the pattern size, then estimated those parameters with iterative optimization [Kennedy and Eberhart 1995]. Recently, GarmentCode [Korosteleva and Sorkine-Hornung 2023] and GarmentCodeData[Korosteleva et al. 2024] presented a procedural way to generate the garment patterns at scale parametrized on body parameters, in which each garment panel is still represented by parametric curves. Design2GarmentCode [Zhou et al. 2025] used fine-tuned Large Multimodal Models to directly generate GarmentCode [Korosteleva and Sorkine-Hornung 2023] programs from multi-modal design input.

Different from the aforementioned works, in this paper, we propose to model a garment pattern as raster data (2D grids), a bitmap representation composed of multiple concepts, including layers, inside/outside flags, edge types, and local deformation matrices [Igarashi et al. 2005; Sorkine and Alexa 2007]. Our work is inspired by methods in computer graphics that map 2D and 3D surfaces to a grid space. Geometry Images represented a 3D surface into a 2D image [Gu et al. 2002], and it has been used to learn 3D surface models by neural networks [Groueix et al. 2018; Sinha et al. 2016]. Similarly, Polycube mapped a 3D surface to a set of 2D grid panels [Tarini et al. 2004] and Polysquare mapped 2D shape onto a 2D grid

[Liu et al. 2017; Xiao et al. 2018]. Due to diverse panel shapes and the equal seam length constraint, we use a simple strategy to robustly project panel shapes onto 2D grids. We also draw inspiration from Shen et al. [2020], which modeled the 3D garment as an image representation within the UV space of the human body. However, their work does not explore pattern representation.

2.2 Garment Pattern Prediction from an Input

Pattern prediction from 3D input. Given a 3D garment mesh as input, the shape is first segmented into patches guided by user sketches [Wang et al. 2005], predefined seams [Bang et al. 2021], or woven fabric properties [Pietroni et al. 2022]. These patches are then parameterized into 2D using surface flattening techniques [Sorkine and Alexa 2007; Wang et al. 2002]. Recently, researchers have explored data-driven approaches for extracting sewing patterns from 3D data [Goto and Umetani 2021; Korosteleva and Lee 2022]. Goto et al. [2021] computed the pattern classification on 3D input from multiple-image segmentation learned by an image translation network [Ronneberger et al. 2015]. NeuralTailor [Korosteleva and Lee 2022] represented a model that converts a 3D point cloud to a garment pattern, learning from a pattern data set [Korosteleva and Lee 2021], where the model first generated a sequence of panels choosing from a set of predefined panel categories and then inferred their shape.

Pattern prediction from 2D image. Given an image as input, a common approach is to match the garment with a parametric sewing pattern, then optimize the pattern's parameters for garment reconstruction from images [Jeong et al. 2015; Su et al. 2020; Yang et al. 2018]. Wang et al. [2018] proposed to learn a shared latent shape space between 2D sketches, garment and body shape parameters, and draped garment shapes by training multiple encoder-decoder networks for each type of garment, enabling fast pattern inference from sketch. Recently, Liu et al. [2023] created a comprehensive dataset with various human poses, body shapes, and sewing patterns, and introduce a two-level Transformer decoder to recover garment panels from learned panel queries defined for all panel types.

Pattern generation from text. Recently, DressCode [He et al. 2024] proposed a method to generate sewing patterns from a text by first quantizing the pattern to a sequence of tokens and using a GPT-based architecture to generate the tokens autoregressively.

These studies represent patterns as discrete panels, either using a set of parameterized Bézier curves or a sequence of tokens. Thus, these approaches encounter the two challenges (discontinuity and less generalizability) outlined in Section 1. In contrast, GarmentImage represents a pattern as integrated raster data, thus avoiding these challenges. We demonstrate its advantages on tasks such as pattern exploration in latent space, text-based pattern editing and image-to-pattern prediction in Section 4.

3 GarmentImage

GarmentImage representation encodes a sewing pattern—including a discrete collection of 2D panels, stitching information, and the placement of each piece on the body—as raster data (Section 3.1). It serves as an intermediate data structure connecting vector-based pattern representation and machine learning models. Given a vector-format garment pattern, the encoding process (Section 3.2) transforms it into a GarmentImage suitable for input to learning-based methods. Conversely, once a GarmentImage is generated by a model, the decoding process (Section 3.3) reconstructs it into a vector-based pattern compatible with existing fashion pipelines, such as simulation. In what follows, we refer to a vector-based sewing pattern simply as a *sewing pattern* or *pattern*, and use the term *GarmentImage* to denote our proposed sewing pattern representation.

3.1 Representation

Figure 2 provides an overview of the four core concepts in the GarmentImage representation. We describe each concept in detail below and explain how their corresponding values are computed in next section (Section 3.2).

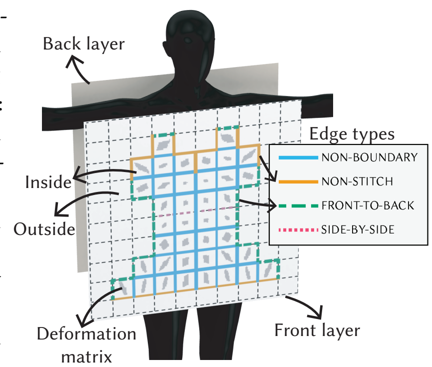

Fig. 2. Representation overview.

3.1.1 Layer. GarmentImage represents a sewing pattern using two distinct layers: a front layer and a back layer. Panels positioned in front of the body are embedded in the front layer, while those behind the body are embedded in the back layer (front and back grid sandwich a T-posed body). Each layer is represented as a 2D array of grid cells. Each grid cell is positioned in a specific location around the 3D human body. Adjacent grid cells are mapped to adjacent regions on the human body surface. Additionally, each grid cell contains an inside/outside flag, four edge types, and a deformation matrix, detailed below.

3.1.2 Inside/Outside Flag. By analogy to parts of the human body covered by a garment panel, we use the inside/outside flag of a cell to indicate whether the cell on the layer is covered by the garment panel or not.

3.1.3 Edge Type. Each grid cell has four edges, and each edge has a type that stores the boundary and stitching information. As illustrated in Figure 2, we define four edge types:

- **NON-BOUNDARY**: The edge is not on a boundary or stitch. Completely inside a panel or outside a panel.
- **NON-STITCH**: The edge is on the boundary of a garment without stitching. This can appear inside a panel with holes.
- **FRONT-TO-BACK**: The edge is on the boundary of a panel and stitched to an edge at the same location on the opposite layer (front-to-back or back-to-front).
- **SIDE-BY-SIDE**: The edge is on the boundary of a panel and stitched to an edge of an adjacent cell on the same layer (front-to-front, back-to-back).

3.1.4 Deformation Matrix. While each grid cell is associated with a local region on a garment panel, the shape of the grid cell (square) is different from the shape of the local region inside the panel (arbitrary quadrilateral). The deformation matrix represents the mapping from the square to the quadrilateral. More specifically, a deformation matrix $F \in \mathbb{R}^{2\times 4}$ consists of four columns, each of which represents a *deformation vector* $f \in \mathbb{R}^{2\times 1}$ of an edge. It is defined as $\bar{v}_j - \bar{v}_i$, where $\bar{v}_i$ is the start of the edge and $\bar{v}_j$ is the end of the edge in the quadrilateral (edges are oriented).

3.2 Encoding

Equipped with the definition of GarmentImage representation described in Section 3.1, the encoding process aims to convert the vector representation of a garment pattern into it.

Given a garment pattern, we first classify the panels into front panels and back panels using the panels' placement information. If a panel covers both the front and back sides of the body, we need to split it into the front part and back part, which are then stitched together along the panel boundary. We then deform the panels so that panel seams are aligned with the corresponding seams to be stitched together (the intermediate layout between Figure 3(a) to (b)). Then, we embed the deformed panels into a grid by discretizing vertex coordinates. Each grid cell is classified as inside if its center lies within one panel, or outside if it lies outside all panels. As we have aligned seam edges between stitched panels before, this discretization can guarantee that (1) panels stitched together is adjacent in the grid space, (2) there is no gap or overlap between panels on the grid, and we can handle stitches between seams with different lengths (see the waistbands in Figure 4(b)).

We then rasterize the front and back grids. After rasterization, GarmentImage can be intuitively considered as a front grid positioned in front of the human body and a back grid positioned behind it. The edge type is automatically determined by the relation between the adjacent cells (Figure 3(b) to (c)): the corresponding FRONT-TO-BACK edges are located in the same position at the front and back grids. A SIDE-BY-SIDE edge is an edge adjacent to another from a neighboring cell on the same grid, where the two cells belong to two switched panels.

The final step is to compute the deformation matrix for each grid cell (Figure 3(b) to (c)). We construct a quad mesh $M = (V, E)$ by connecting grid cells associated with a panel, where V is mesh vertices and E is the mesh edges. We then deform the quad mesh M to $\bar{M} = (\bar{V}, \bar{E})$ so that its boundary matches the boundary of the original panel while minimizing the edge deformation. We formulate this as a least squares problem with a linear constraint Equation 1:

$$\arg\min_{\bar{v}}\{\sum_{(i,j)\in E}\left((\bar{v}_j - \bar{v}_i) - (v_j - v_i)\right)^2\} \quad \text{s.t.} \quad C\bar{v} = c \tag{1}$$

where v_i and $\bar{v}_i$ are the positions of the i-th vertex of M before and after deformation, $(i, j) \in E$ indicates a directed edge in M from vertex v_i to vertex v_j. The matrix C encodes the linear constraint that enforces the boundary vertices in $\bar{M}$ to match their target positions on the panel boundary curve, given by c. We solve Equation 1 using Lagrange multipliers [GOLUB 2005]. The deformation vector is defined as $\bar{v}_j - \bar{v}_i$, and the deformation matrix for a cell is formed by stacking four such deformation vectors as columns.

3.3 Decoding

Although deformation and discretization squash or stretch the original panel during encoding, the decoding process aims at reconstructing the sewing pattern (panels, stitches, and placement of panels) from GarmentImage that might be generated by a machine learning model (Figure 3(c) to (d)). This process is deterministic and fully automated. We compute it in three steps as follows.

Getting panels by cell clustering. First, different panels need to be separated from GarmentImage. We group grid cells flagged as inside and enclosed by boundary edges (i.e., grid cell edges with edge type NON-STITCH, FRONT-TO-BACK, SIDE-BY-SIDE). Each resulting group of grid cells corresponds to a distinct garment panel.

Panel shape recovery. Next, for each identified panel, we aim to recover its shape. We construct a quad mesh $M = (V, E)$ from the group of grid cells as that of in encoding, and deform the quad mesh M to $\hat{M} = (\hat{V}, \hat{E})$. The deformation aims to minimize the difference between resulting edge $\hat{e}_{i,j} = \hat{v}_j - \hat{v}_i$, and embedded deformation vector $f_{i,j}$ while keeping the location of the resulting mesh on the grid. We formulate it as another least squares problem:

$$\arg\min_{\hat{v}}\{\sum_{i,j\in E}((\hat{v}_j - \hat{v}_i) - f_{ij})^2 + \sum_{v_i\in V}(\hat{v}_i - v_i)^2\} \tag{2}$$

where v_i is the vertex position in M, $(i, j) \in E$ indicates a directed edge in M from vertex v_i to vertex v_j and f_{ij} is its encoded deformation vector.

Stitching and placement recovery. With the recovered panel shape, we extract stitching information by transferring edge types from the 2D grids to their corresponding deformed edges on the panel curves. This defines how different panels are connected. Additionally, the location of the grid cells associated with a panel indicates the panel's intended placement around the human body. This spatial information enables the reconstruction of a 3D garment mesh around a 3D human model for physical simulation, or the generation of 2D printable patterns that can be cut and stitched to produce a physical garment.

3.4 Examples

In Figure 4, we show that our representation can represent a large variety of sewing patterns and design features such as waistbands and darts. Additionally, GarmentImage can naturally represent panels with holes, a capability that would require a dedicated template [Korosteleva and Lee 2021, 2022] or command [Korosteleva et al. 2024] in vector-based representation.

4 Experiments

In this section, we introduce three downstream tasks to illustrate the advantages of GarmentImage. Across all experiments, we use a 16×16 GarmentImage representation with 34 channels per grid cell, resulting in a shape of $(34, 16, 16)$. Of these 34 channels, 17 are allocated to the front layer and the remaining 17 to the back layer. Each set of 17 channels includes one channel for the inside/outside flag, eight channels encoding the edge types for the bottom and left edges (as two one-hot vectors of length four), and eight channels representing the deformation matrix. To build our experimental datasets, we first sampled diverse sewing patterns by randomizing values of

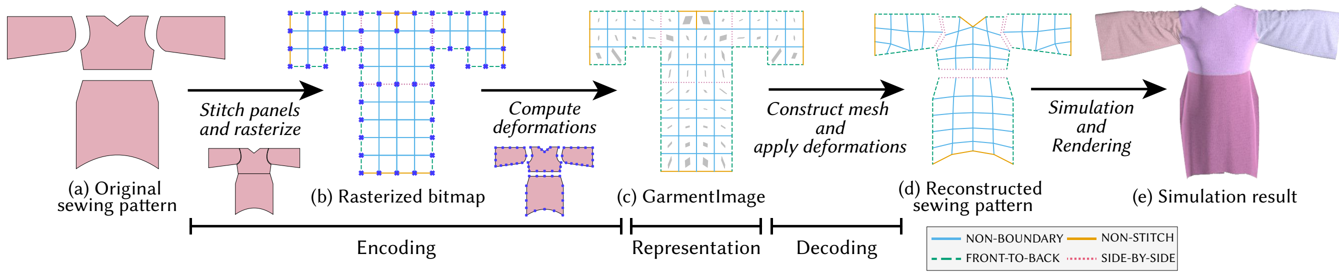

Fig. 3. **GarmentImage encoding and decoding process.** A GarmentImage is automatically encoded from a sewing pattern in vector format and can be decoded back to the vector format. Given a sewing pattern in vector format (a), we stitch the panels and fill the gaps between them before rasterizing it into a bitmap (b). During this process, we establish correspondences between the original panel curves and the vertices on the bitmap grid. This allows us to assign edge types to each grid edge and compute deformation matrices that align the grid cells with the original panels. The resulting GarmentImage representation for each cell (c) contains inside/outside flags, edge types, and a deformation matrix, visualized as a parallelogram in the cell. For the decoding process, starting from the GarmentImage, we reconstruct the sewing pattern in vector format (d), which can then be used for applications such as simulation (e).

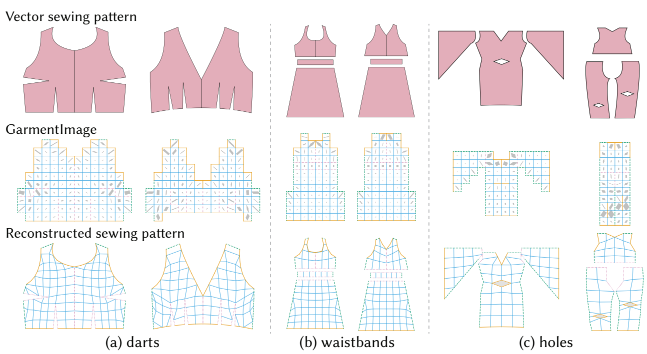

Fig. 4. **GarmentImage examples.** GarmentImage can handle diverse garments and features such as darts (a), waistbands (b), and holes (c).

various parameters (e.g., sleeve length) defined in sewing pattern dataset Korosteleva *et al.* [2021]. Corresponding GarmentImages representations were then automatically generated from these patterns by the encoding process introducted in Section 3.2. In addition, we employed the simulator proposed in Korosteleva *et al.* [2024] to collect simulated images from sewing patterns.

4.1 VAE Latent Space Exploration

We hypothesize that GarmentImage enhances the robustness of a trained VAE to new topologies, leading to a more continuous latent space than that of vector-based representation. To validate this, we train a VAE using GarmentImage (dubbed GarmentImage VAE) and another using a vector-based pattern representation (vector-based VAE) on the same reconstruction task. We compare their performance based on latent space interpolation and extrapolation results.

Since GarmentImage is a raster format, we can take advantage of popular neural network architectures such as convolutional neural networks (CNNs), which are well-suited for raster data. We train a CNN-based VAE with 32 dimension latent space using GarmentImage. For the vector-based representation, we adopt the similar pattern format proposed in [Korosteleva and Lee 2021]. It consists of $4 \times n \times 16$ dimensions, where 4 corresponds to (x, y, curvature x, curvature y), n is the number of panel types, and 16 is the maximum number of vertex in a panel. To process this format, we employ a transformer-based model like Sewformer [Liu et al. 2023]. Specifically, we use a transformer-based VAE architecture inspired by Motionformer [Petrovich et al. 2021], which injects two special tokens into the input sequence to predict the mean μ and standard deviation σ of the latent presentation. See Figure 8 for more details.

We prepare three datasets. The first dataset (Figure 9(a,b)) includes three garment pattern types–*dresses*, *jumpsuits*, and *top + pants*, and serves to assess whether the trained VAE can generate an entirely new garment pattern with new topology (*two-panel dresses*) through latent-space exploration. The second dataset (Figure 9(c,d)) contains a wider variety of patterns, such as *one-panel dress with sleeves* and *top + skirt with sleeves*, and is used to evaluate the smoothness of the latent spaces when performing latent space interpolation across diverse garment pattern types. The third dataset (Figure 9(e, f)) consists of *one-panel shirts* and *two-panel shirts with* and *without darts*, and is used to test latent space interpolation of GarmentImage in the presence of the complicated garment feature. We collected around 20,000 garment sewing patterns and GarmentImages for each garment type.

Latent space interpolation. As shown in Figure 9(a,c,e), the latent space of GarmentImage VAE exhibits continuous interpolations between garments of different topologies. For example, when interpolating between a *top + pants* and a *one-panel dress* pattern (Figure 9(a)), the topology of generated pattern transitions continuously in the GarmentImage VAE. In contrast, the vector-based VAE produces discrete jumps, often resulting in invalid garment patterns—an undesirable behavior for latent space interpolation. Additionally, our method can also support a continuous interpolation in the number and size of darts, as shown in Figure 9(e).

Latent space extrapolation. We assess the model's ability to generate patterns with unseen topologies through the latent space extrapolation experiments. As shown in Figure 9(b), when we aim to perform a topology edit – applying the latent vector difference from a *jumpsuit* to *top + pants* patterns to a *dress*, GarmentImage VAE successfully splits the *dress* into top and skirt panels, while the

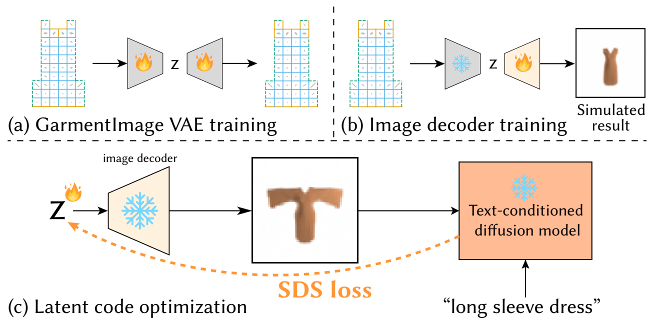

Fig. 5. **Text-based pattern editing pipeline.** (a) We first train a GarmentImage VAE encoder and decoder on the GarmentImage reconstruction task. (b) Then we train an image decoder that predicts a simulated result in 64×64 given a latent code from GarmentImage VAE. (c) We optimize for the best GarmentImage VAE latent code that minimizes the SDS loss [Poole et al. 2023] to conform to the input text prompt.

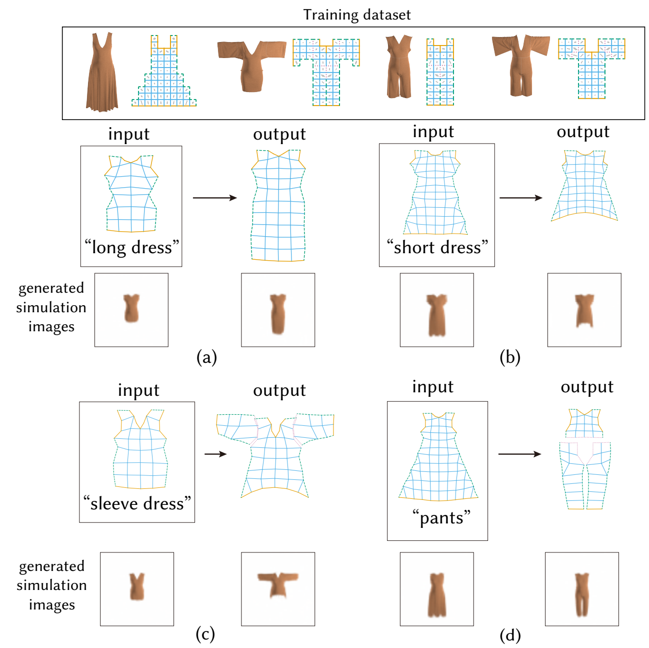

Fig. 6. **Text-based pattern editing results.** The input GarmentImage patterns adjust their geometry (a, b) and even topology (c, d) to match the input text prompts.

vector-based VAE fails to capture this transfer. In Figure 9(d), GarmentImage VAE can even generate an unseen pattern (*top + skirt*) purely by the topology edit in the latent space. In Figure 9(f), GarmentImage VAE successfully transfers the adding darts edit. These findings demonstrate that GarmentImage VAE's latent space remains continuous and robust to topological changes, even for garments not encountered during training.

4.2 Text-based Pattern Editing

We demonstrate an optimization-based text-based pattern editing application, using the GarmentImage VAE latent space described in Section 4.1. In Figure 5, we optimize the latent code within this space to align with a given text prompt. We use Stable Diffusion[1] [Rombach et al. 2022] as the text-to-image generator. To bridge the domain gap with Stable Diffusion, we employ an image decoder that maps the VAE latent code to a simulated garment image. Finally, we utilize the SDS loss [Poole et al. 2023] to minimize the distribution gap in the diffusion model's latent space between the generated simulation image and the target text prompt. We collected a dataset with paired GarmentImage and simulation result image for four garment types: *one-panel sleeveless dresses*, *one-panel dresses with sleeves*, *tops + pants*, and *tops with sleeves + pants*. For simplicity, all simulated garments are rendered from the front view in brown, at a resolution of 64×64. We train the image decoder on these paired samples while keeping the VAE encoder fixed. As shown in Figure 6, the initial GarmentImages adapt their shapes and topologies to match the target prompts, thanks to the continuous latent space. For instance, in Figure 6(d), the *one-panel dress* transforms into a *top + pants* garment, while preserving the shape of the original top's silhouette, based on the text prompt "pants".

4.3 Image-to-Pattern Prediction

In this experiment, we compare a model based on GarmentImage with Sewformer [Liu et al. 2023] and NeuralTailor [Korosteleva and Lee 2022] on image-to-pattern prediction task. Our goal is to demonstrate the generalization advantages of our representation against new garment topologies.

4.3.1 *Network Architecture.*

Figure 10 illustrates our use of a simple U-Net model to sequentially predict a GarmentImage. First, the model predicts inside/outside flags from the input image. Next, it predicts edge types using both the input image and the previously predicted flags. Finally, the deformation matrix is predicted based on the input image, inside/outside flags, and edge types. We use L2 loss for the inside/outside flag and deformation matrix prediction, and cross-entropy loss for edge type prediction.

4.3.2 *Training and Evaluation.*

We evaluate the model's generalizability to unseen garment types in two experimental settings.

Generalizability to an entirely new topology. In this experiment, we evaluate whether the trained models can generalize to a garment topology that was never encountered during training (Figure 11 left). To this end, we constructed a dataset D1 including four garment types: *one-panel dresses*, *one-panel jumpsuits*, *top + pants*, and *top + skirt*. Each entry in D1 includes a sewing pattern, its GarmentImage representation and simulated image. The top panels are rendered in light blue, while the others are brown, clearly indicating the separation between the top and bottom panels. We train the model using only *dresses*, *jumpsuits*, and *top + pants* garments. We test the model on *top + skirt* garments to assess how well the model generalizes to this entirely new topology. We collected approximately 80, 000 garments for training. We compare our approach with Sewformer [Liu et al. 2023]. Figure 11 (left) shows that our model accurately predicts separate skirt panels for the garment's bottom, while Sewformer is

[1] https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5

Table 1. We report the IoU values between the predicted sewing patterns and ground truth for both seen and unseen panel combinations on dataset D2. For seen panel combinations, our approach outperforms NeuralTailor [Korosteleva and Lee 2022] by 9.1% and achieves performance comparable to Sewformer [Liu et al. 2023]. Notably, for the unseen panel combination, our approach significantly surpasses both baseline methods.

	Seen combinations	Unseen combination
NeuralTailor	0.8390	0.3131
Sewformer	**0.9311**	0.4260
Ours	0.9304	**0.8127**

limited to generating only pants panels or invalid ones. We attribute Sewformer's limitations to its neural network architecture, designed to output vector-based sewing pattern representations. For each garment type, Sewformer implicitly performs panel selection from predefined panel pools, restricting its flexibility and making it difficult to generalize to entirely new garment topologies. In contrast, GarmentImage encodes garment topology through a combination of layers, inside/outside flags and edge types, enabling it to handle new garment configurations without needing prior definitions.

Generalizability to an unseen panel combination. In this experiment, we investigate whether the trained model can generalize to a garment panel combination not presented in the training data (Figure 11 right). To this end, we constructed a dataset D2 with four garment types, *one-panel dresses*, *dresses with sleeves*, *top + pants*, and *top with sleeves + pants*. To distinguish pants from skirts, we render the pants in light blue. We train the model using only *one-panel dresses*, *dresses with sleeves*, and *top + pants* garments. And evaluate on *top with sleeves + pants* garments—an unseen combination of panels. Note that while each individual panel (top, sleeves, and pants) is present in the training data, their combination in this form is not. We compare our approach with Sewformer [Liu et al. 2023]. Our training dataset contains approximately 80, 000 garments, while the test set includes around 6, 500 garments of seen types and 3, 500 of the unseen type. Figure 11 (right) shows that our model correctly predicts the new panel combination, while Sewformer fails to generate the sleeves and even valid patterns.

We also compute the intersection-over-union (IoU) between the predicted sewing patterns and ground truth patterns for each panel aligning their centroids for a fair comparison. Table 1 presents the results alongside NeuralTailor [Korosteleva and Lee 2022] and Sewformer. For NeuralTailor, we adapt the model to accept image inputs by replacing its point cloud encoder with a pre-trained ResNet-50 [He et al. 2016]. This modified encoder extracts both per-pixel features and a global image representation, which are then fed into NeuralTailor's original decoder to generate sewing patterns. The results indicate that our model demonstrates superior generalizability to the unseen panel combination compared to both NeuralTailor and Sewformer.

5 Limitations and Future Work

While our method performs well in several applications, we acknowledge that there is significant room for improvement before it can be widely embraced by the fashion industry.

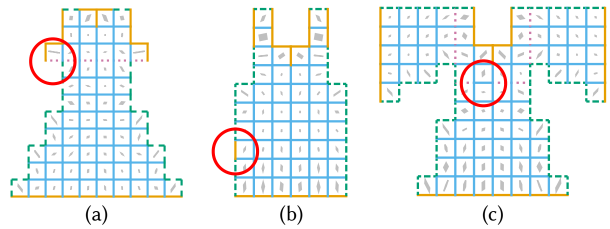

Fig. 7. (a) **Invalid encoded GarmentImage.** Our automatic encoding process may generate invalid GarmentImage representations. In this case, the grid cell containing the SIDE-BY-SIDE edge should be marked as inside, but it is not. (b, c) **Invalid predicted GarmentImage.** Our proposed neural network may predict invalid GarmentImage representations. In (b), the NON-STITCH edge should be predicted as FRONT-TO-BACK, and in (c), the NON-BOUNDARY edge should be predicted as SIDE-BY-SIDE.

Non-smooth pattern boundaries. A reconstructed sewing pattern from GarmentImage may appear distorted or less smooth compared to its original vector-based representation. This limitation can be mitigated by increasing the resolution of GarmentImage and incorporating post-processing to smooth the reconstructed patterns.

Non-uniqueness of the representation. The same pattern can be encoded into different GarmentImages. For example, a panel may be represented as either a dense arrangement of many small cells or a sparse configuration of fewer large cells. To minimize inconsistencies and potential negative impacts on model performance, it is important to follow a consistent policy when converting patterns into GarmentImages for training.

Limitation of the automatic encoding. In our experiments, we constructed the GarmentImage datasets by automatically encoding vector-based patterns. However, the current automatic encoding process is limited to simpler garment types, such as *dresses* and *jumpsuits*, and may produce invalid GarmentImages (see Figure 7(a)). The success rate for automatic encoding is 89.88% for *dresses* in the NeuralTailor dataset [Korosteleva and Lee 2022]. We filtered out the unsuccessful examples from the training dataset using simple filtering rules, without requiring human intervention. Handling more complex garments and addressing corner cases may require further processing or manual annotation.

Invalid predicted GarmentImage result. GarmentImage representations generated by neural networks might yield invalid garment patterns after decoding. We occasionally encounter invalid outputs, often due to incorrect edge type predictions. We automatically detect two issues: a NON-STITCH edge between two FRONT-TO-BACK edges (Figure 7(b)), and a NON-BOUNDARY edge between two SIDE-BY-SIDE edges (Figure 7(c)). These errors are corrected by reassigning the appropriate edge types. However, fully resolving more complex inconsistencies may require human intervention.

Unrealistic garment dataset. Currently, the experiments in Section 4.2 and Section 4.3 utilize front-view renderings of simulated garments. While this setup effectively showcases the superior generalizability of our method, incorporating more realistic images will be essential for future work aimed at practical, real-world applications.

Extend GarmentImage to more diverse garment features. In this work, we introduce the basic GarmentImage, which comprises two layers corresponding to the front and back sides of a garment. While layered garment features such as collars, cuffs, and pockets can be defined as separate panels on the new layer, tuck requires a distinct edge type labelled as **TUCK**.

6 Conclusion

In this work, we presented GarmentImage, a novel raster representation for diverse garment sewing patterns. GarmentImage encodes geometry, topology, and placement information into multi-channel grids, providing a unified alternative to traditional vector-based pattern representations. We demonstrated the advantages of GarmentImage using three applications: VAE latent space exploration, text-based pattern editing and image-to-pattern prediction. Our results show that models trained with GarmentImage exhibit a more continuous latent space and improved generalization to unseen topologies, compared to vector-based pattern representations. Looking further, we hope that our work will inspire research exploring the use of the proposed representation in various garment design applications.

Acknowledgments

We thank Nicolas Rosset for the advice on VAE experiments and the anonymous reviewers for their valuable feedback. This work was supported in part by the Japan Science and Technology Agency (JST) as part of Adopting Sustainable Partnerships for Innovative Research Ecosystem (ASPIRE), Grant Number JPMJAP2401, and JSPS Grant-in-Aid JP23K16921, Japan.

References

Winifred Aldrich. 2015. *Metric pattern cutting for women's wear.* John Wiley & Sons.

Helen Joseph Armstrong. 2014. *Patternmaking for fashion design.* Pearson.

Seungbae Bang, Maria Korosteleva, and Sung-Hee Lee. 2021. Estimating garment patterns from static scan data. In *Computer Graphics Forum*, Vol. 40. 273–287.

Aric Bartle, Alla Sheffer, Vladimir G Kim, Danny M Kaufman, Nicholas Vining, and Floraine Berthouzoz. 2016. Physics-driven pattern adjustment for direct 3D garment editing. *ACM Trans. Graph.* 35, 4 (2016), 50–1.

Remi Brouet, Alla Sheffer, Laurence Boissieux, and Marie-Paule Cani. 2012. Design preserving garment transfer. *ACM Transactions on Graphics* 31, 4 (2012), Article–No.

Cheng-Hsiu Chen, Jheng-Wei Su, Min-Chun Hu, Chih-Yuan Yao, and Hung-Kuo Chu. 2024. Panelformer: Sewing Pattern Reconstruction from 2D Garment Images. In *Proceedings of the IEEE/CVF WACV.* 454–463.

GENE H GOLUB. 2005. CME 302: NUMERICAL LINEAR ALGEBRA FALL 2005/06 LECTURE 10. (2005).

Chihiro Goto and Nobuyuki Umetani. 2021. Data-driven Garment Pattern Estimation from 3D Geometries.. In *Eurographics (Short Papers).* 17–20.

Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 2018. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE CVPR.* 216–224.

Xianfeng Gu, Steven J Gortler, and Hugues Hoppe. 2002. Geometry images. *ACM Trans. Graph.*, 355–361.

Kai He, Kaixin Yao, Qixuan Zhang, Jingyi Yu, Lingjie Liu, and Lan Xu. 2024. Dresscode: Autoregressively sewing and generating garments from text guidance. *ACM Trans. Graph.* 43, 4 (2024), 1–13.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE CVPR.* 770–778.

Takeo Igarashi, Tomer Moscovich, and John F. Hughes. 2005. As-Rigid-as-Possible Shape Manipulation. *ACM Trans. Graph.* 24, 3 (jul 2005), 1134–1141.

Moon-Hwan Jeong, Dong-Hoon Han, and Hyeong-Seok Ko. 2015. Garment capture from a photograph. *Computer Animation and Virtual Worlds* 26, 3-4 (2015), 291–300.

James Kennedy and Russell Eberhart. 1995. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, Vol. 4. ieee, 1942–1948.

Maria Korosteleva, Timur Levent Kesdogan, Fabian Kemper, Stephan Wenninger, Jasmin Koller, Yuhan Zhang, Mario Botsch, and Olga Sorkine-Hornung. 2024. GarmentCodeData: A dataset of 3D made-to-measure garments with sewing patterns. In *European Conference on Computer Vision.* Springer, 110–127.

Maria Korosteleva and Sung-Hee Lee. 2021. Generating Datasets of 3D Garments with Sewing Patterns. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, Vol. 1.

Maria Korosteleva and Sung-Hee Lee. 2022. NeuralTailor: Reconstructing Sewing Pattern Structures from 3D Point Clouds of Garments. *ACM Trans. Graph.* 41, 4, Article 158 (jul 2022), 16 pages. https://doi.org/10.1145/3528223.3530179

Maria Korosteleva and Olga Sorkine-Hornung. 2023. GarmentCode: Programming Parametric Sewing Patterns. *ACM Trans. Graph.* 42, 6 (2023), 1–15.

Minchen Li, Alla Sheffer, Eitan Grinspun, and Nicholas Vining. 2018. Foldsketch: Enriching garments with physically reproducible folds. *ACM Trans. Graph.* 37, 4 (2018), 1–13.

Celong Liu, Wuyi Yu, Zhonggui Chen, and Xin Li. 2017. Distributed poly-square mapping for large-scale semi-structured quad mesh generation. *Computer-Aided Design* 90 (2017), 5–17.

Lijuan Liu, Xiangyu Xu, Zhijie Lin, Jiabin Liang, and Shuicheng Yan. 2023. Towards garment sewing pattern reconstruction from a single image. *ACM Trans. Graph.* 42, 6 (2023), 1–15.

Rahul Narain, Armin Samii, and James F O'brien. 2012. Adaptive anisotropic remeshing for cloth simulation. *ACM Trans. Graph.* 31, 6 (2012), 1–10.

Mathis Petrovich, Michael J Black, and Gül Varol. 2021. Action-Conditioned 3D Human Motion Synthesis with Transformer VAE. In *2021 IEEE ICCV.*

Nico Pietroni, Corentin Dumery, Raphael Falque, Mark Liu, Teresa Vidal-Calleja, and Olga Sorkine-Hornung. 2022. Computational Pattern Making from 3D Garment Models. *ACM Trans. Graph.* 41, 4 (2022), 157:1–14.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2023. DreamFusion: Text-to-3D using 2D Diffusion. In *ICLR.*

Anran Qi and Takeo Igarashi. 2024. PerfectTailor: Scale-Preserving 2D Pattern Adjustment Driven by 3D Garment Editing. *IEEE CG&A* (2024).

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF CVPR.* 10684–10695.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI 2015.* Springer, 234–241.

Yu Shen, Junbang Liang, and Ming C Lin. 2020. Gan-based garment generation using sewing pattern images. In *ECCV.* Springer, 225–247.

Ayan Sinha, Jing Bai, and Karthik Ramani. 2016. Deep learning 3D shape surfaces using geometry images. In *ECCV.* Springer, 223–240.

Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, Vol. 4. Citeseer, 109–116.

Zhaoqi Su, Weilin Wan, Tao Yu, Lingjie Liu, Lu Fang, Wenping Wang, and Yebin Liu. 2020. Mulaycap: Multi-layer human performance capture using a monocular video camera. *IEEE TVCG* 28, 4 (2020), 1862–1879.

Marco Tarini, Kai Hormann, Paolo Cignoni, and Claudio Montani. 2004. Polycube-maps. *ACM Trans. Graph.* 23, 3 (2004), 853–860.

Charlie CL Wang, Shana SF Smith, and Matthew MF Yuen. 2002. Surface flattening based on energy model. *Computer-Aided Design* 34, 11 (2002), 823–833.

Charlie CL Wang, Yu Wang, and Matthew MF Yuen. 2005. Design automation for customized apparel products. *Computer-aided design* 37, 7 (2005), 675–691.

Huamin Wang. 2018. Rule-free sewing pattern adjustment with precision and efficiency. *ACM Trans. Graph.* 37, 4 (2018), 1–13.

Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popovic, and Niloy J. Mitra. 2018. Learning a Shared Shape Space for Multimodal Garment Design. *ACM Trans. Graph.* 37, 6 (2018), 1:1–1:14. https://doi.org/10.1145/3272127.3275074

Shiwei Xiao, Hongmei Kang, Xiao-Ming Fu, and Falai Chen. 2018. Computing IGA-suitable planar parameterizations by PolySquare-enhanced domain partition. *Computer Aided Geometric Design* 62 (2018), 29–43.

Shan Yang, Zherong Pan, Tanya Amert, Ke Wang, Licheng Yu, Tamara Berg, and Ming C Lin. 2018. Physics-inspired garment recovery from a single-view image. *ACM Trans. Graph.* 37, 5 (2018), 1–14.

Feng Zhou, Ruiyang Liu, Chen Liu, Gaofeng He, Yong-Lu Li, Xiaogang Jin, and Huamin Wang. 2025. Design2GarmentCode: Turning Design Concepts to Tangible Garments Through Program Synthesis. In *Proceedings of the IEEE/CVF CVPR.*

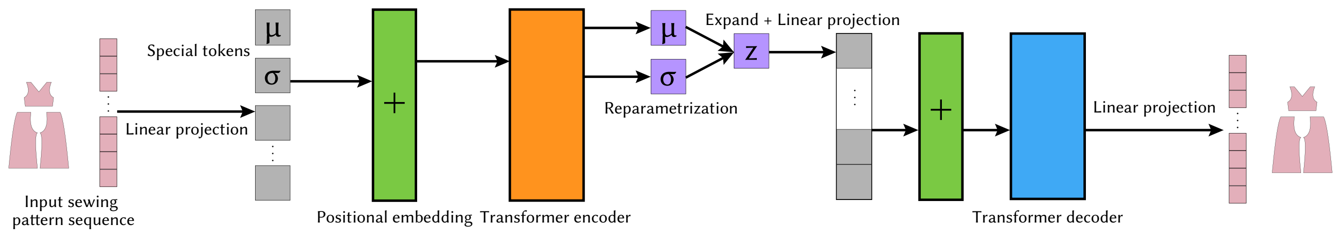

Fig. 8. **Network architecture of vector-based VAE.** We employ a transformer-based architecture that takes a sewing pattern sequence and two special tokens (μ and σ tokens) as input. Specifically, we give the input tokens into the transformer encoder to produce the meanμ and standard deviation σ of the latent presentation. We then compute z by reparametrization trick and pass it into the transformer decoder to reconstruct the input sewing pattern sequence. Note that this VAE only predicts the panels' shape and does not output stitch information.

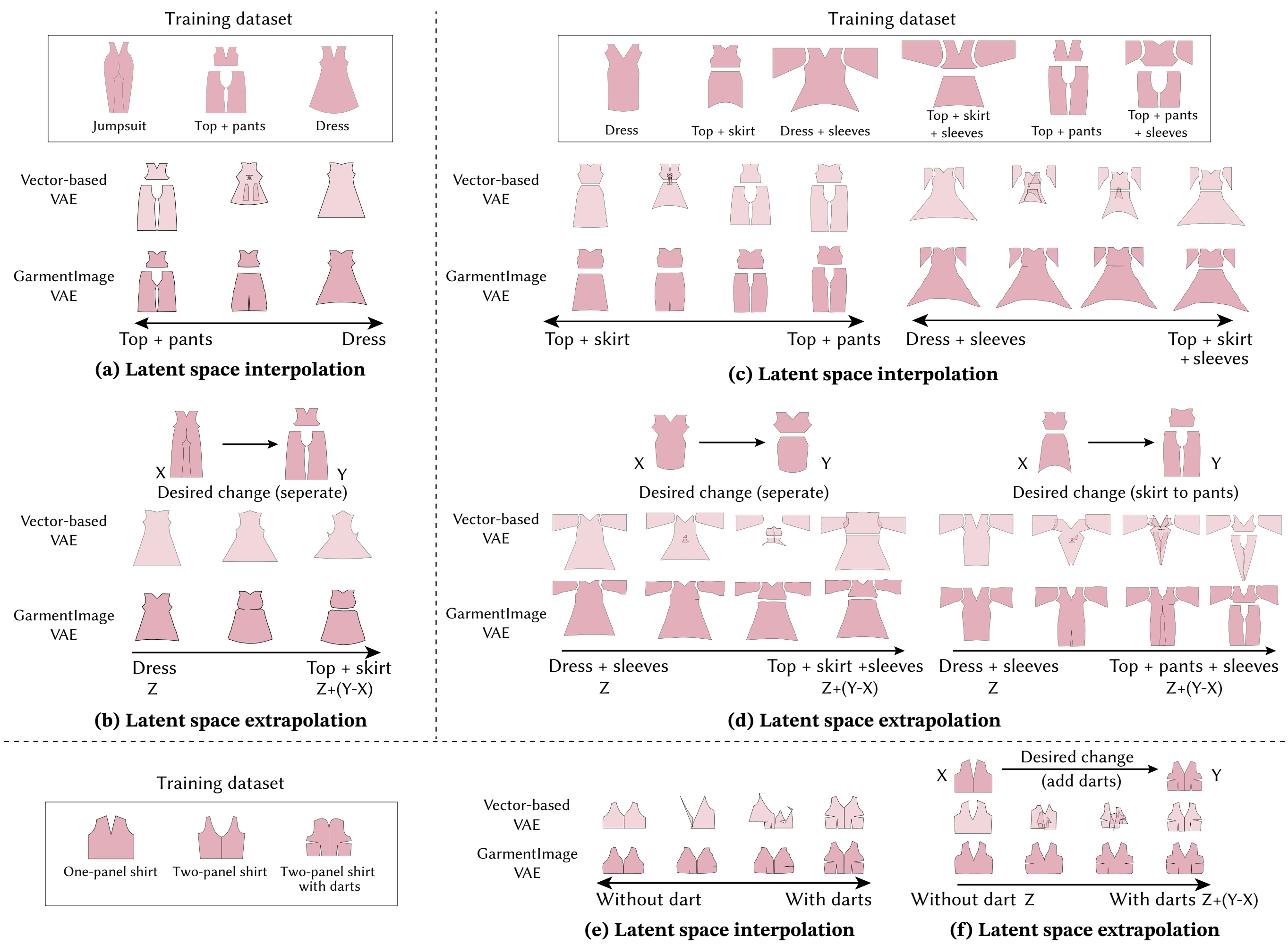

Fig. 9. **Latent space experiment results.** The latent space of GarmentImage VAE showcases more continuous interpolations and extrapolations between different topologies than that of vector-based VAE. In (c, left), when interpolating a *Top + skirt* into *Top + pants* pattern, GarmentImage VAE continuously splits the skirt panel into two panels for pants, whereas the vector-based VAE exhibits discrete jumps. A similar behavior is observed in (a). In (e), our method achieves a continuous interpolation in the number and size of darts, while the vector-based VAE fails to generate valid dart designs. In (b), we demonstrate a topology edit by transferring the latent vector difference from a *one-panel jumpsuit* to *top + pants* pattern, and applying it to a *one-panel dress*. GarmentImage VAE successfully transfers the edit, splitting the *one-panel dress* pattern into top and skirt panels. In contrast, the vector-based VAE fails to capture the intended transfer. Furthermore, in (d) GarmentImage VAE can generate a previously unseen pattern—a *two-panel dress*—purely by latent space extrapolation. In (f), GarmentImage VAE successfully transfers the adding darts edit onto a *one-panel shirt*. The vector-based VAE fails in both cases.

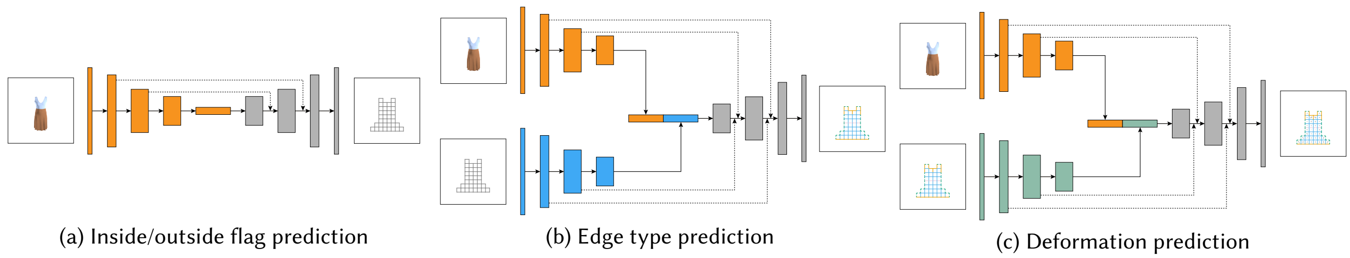

Fig. 10. **Network architecture for Image-to-Pattern prediction (Section 4.3).** We employ three separate U-Net models in a step-by-step pipeline. Given an input image, (a) the first U-Net predicts the inside/outside flags. (b) Using the same input image and the predicted inside/outside flags, the second U-Net infers the edge types. Finally, (c) the third U-Net predicts deformation matrices from the input image and the previously generated inside/outside flags and edge types. Each model is trained independently using its ground truth intermediate representations. During inference, these models are applied sequentially.

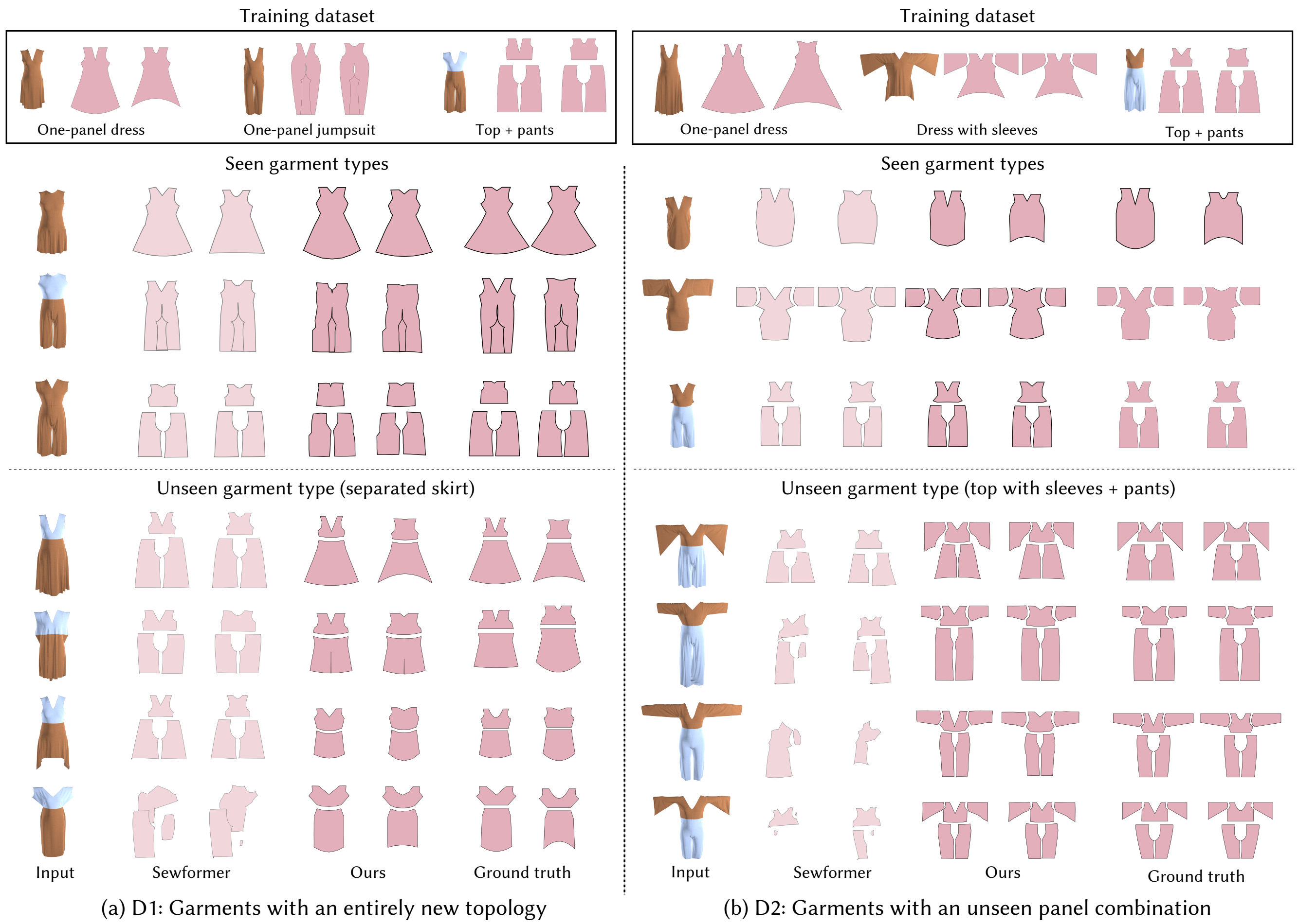

Fig. 11. **Image-to-Pattern prediction results (Section 4.3)**. We visualize the predicted patterns from an input image using our method and Sewformer [Liu et al. 2023]. While both methods can predict reasonable sewing patterns for garment types included in the training dataset (top), our method demonstrates superior generalizability for unseen garment types (bottom). For garments with an entirely new topology (a), our method predicts accurate patterns, whereas Sewformer usually defaults to patterns presented in the training data. For garments with unseen panel combination (b), Sewformer often produces invalid patterns, and it frequently omits sleeve panels, despite that the sleeve panel is clearly included in *dress with sleeves* in the training dataset. In contrast, our method accurately predicts the correct garment patterns.