GarmentDiffusion: 3D Garment Sewing Pattern Generation with Multimodal Diffusion Transformers

Xinyu Li[1,2] , **Qi Yao**[2] , **Yuanda Wang**[2]
[1]Zhejiang University
[2]Shenfu Research
lixinyu0801@zju.edu.cn, {yaoqi, adamwang}@dejaai.com

Abstract

Garment sewing patterns are fundamental design elements that bridge the gap between design concepts and practical manufacturing. The generative modeling of sewing patterns is crucial for creating diversified garments. However, existing approaches are limited either by reliance on a single input modality or by suboptimal generation efficiency. In this work, we present ***GarmentDiffusion***, a new generative model capable of producing centimeter-precise, vectorized 3D sewing patterns from multimodal inputs (text, image, and incomplete sewing pattern). Our method efficiently encodes 3D sewing pattern parameters into compact edge token representations, achieving a sequence length that is **10**× shorter than that of the autoregressive SewingGPT in DressCode. By employing a diffusion transformer, we simultaneously denoise all edge tokens along the temporal axis, while maintaining a constant number of denoising steps regardless of dataset-specific edge and panel statistics. With all combination of designs of our model, the sewing pattern generation speed is accelerated by **100**× compared to SewingGPT. We achieve new state-of-the-art results on DressCodeData, as well as on the largest sewing pattern dataset, namely GarmentCodeData. The project website is available at *https://shenfu-research.github.io/Garment-Diffusion/*.

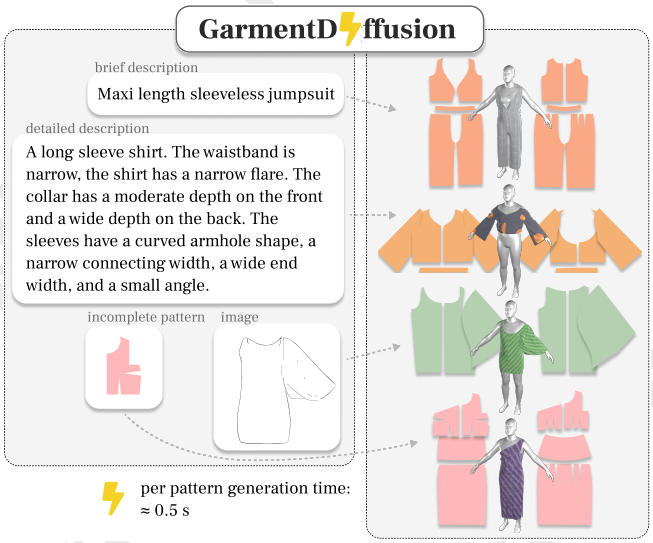


Figure 1: **3D garment pattern generation with multimodal inputs.** As illustrated on the left, our model supports various input modalities, including brief and detailed text descriptions, images (garment sketches), and incomplete patterns. The generated patterns can be draped on human models and utilized for practical production. Our model supports both simple and complex pattern generation, e.g., DressCodeData and GarmentCodeData. As indicated by the **"lightning"** icon, our model demonstrates an ultra-fast pattern generation speed (within a second using a single A10 GPU), which is comparable to the discriminative model (SewFormer).

1 Introduction

Digital garment modeling has emerged as a pivotal research area for fashion design, with garment sewing patterns being essential in transforming design concepts into tangible garments. Many studies have been conducted on sewing pattern modeling and generation, either by scaling up the size of datasets [Korosteleva and Lee, 2021; Korosteleva *et al.*, 2024] or by proposing new parametric and learning-based approaches [Korosteleva and Lee, 2022; Korosteleva and Sorkine-Hornung, 2023; Liu *et al.*, 2023; Chen *et al.*, 2024a; He *et al.*, 2024].

The first attempt [Korosteleva and Lee, 2021] was made to build a synthetic garment sewing pattern dataset using a parametric approach, with ˜20K sewing patterns. However, both the complexity of sewing pattern geometries and the quantity of sewing patterns are insufficient to meet the increasing data demands of advanced data-driven models. Subsequently, in [Korosteleva *et al.*, 2024], the authors built a large-scale garment dataset, i.e., GarmentCodeData, using component-oriented garment programs [Korosteleva and Sorkine-Hornung, 2023]. It introduces more complex design prototypes and scales up the number of sewing patterns by 5×, with ˜115K sewing patterns in total. However, learning-based models typically require paired samples for conditional training, such as `(sewing pattern, modality-specific input)`. The unimodal nature of these datasets restricts the development of generative mod-

eling approaches. SewFactory [Liu *et al.*, 2023] has recognized this problem and provides approximately 1M image-and-sewing-pattern pairs for training. Since its design prototypes are derived from [Korosteleva and Lee, 2021], the complexity of sewing pattern geometry is still limited.

Another challenge lies in efficiently modeling the generation of sewing patterns to make it more applicable to real-time scenarios. SewFormer [Liu *et al.*, 2023] introduces a DETR-like [Carion *et al.*, 2020] discriminative model to map 2D images to sewing patterns. The discriminative training of SewFormer leads to the *deterministic* predictions of sewing patterns given input images, which restricts the diversity of garment designs compared to the generative approaches. A pioneering work, DressCode [He *et al.*, 2024], introduces a GPT-like autoregressive model (SewingGPT) to generate vector-quantized sewing patterns, conditioned on the text descriptions via cross-attention. While this method is effective in generating simple sewing patterns, it faces significant challenges when applied to GarmentCodeData [Korosteleva *et al.*, 2024]. For example, the token sequence length of SewingGPT increases from ˜2K to over 18K, making its training and inference infeasible in practice. Another issue is the coarse sewing pattern descriptions generated by GPT-4V using its data annotation pipeline, which lacks the precise control for text-conditioned sewing pattern generation.

In this paper, we rethink the modeling paradigm of sewing patterns, questioning whether the vector-quantized encoding scheme and autoregressive *next-parameter prediction* in DressCode are efficient for sewing pattern generation. Inspired by BRepGen [Xu *et al.*, 2024], we encode edge-related parameters (such as 3D coordinates, stitch tags, and free edge scores) into the **embedding dimension**, while denoising all edge tokens in parallel along the temporal axis. By leveraging the parallel processing nature of diffusion transformers [Peebles and Xie, 2023], our approach accelerates the generation process by approximately a hundredfold without sacrificing the precision (in centimeters) of sewing pattern geometries. Specifically, with parameters set to `#edge parameters/edge = 9` (endpoints, control points, arc), `#edges/panel = 39`, and `#panels/pattern = 37`, SewingGPT requires $18,135 + 2$(SOS,EOS) tokens and steps to autoregressively generate a sewing pattern. In contrast, our model only needs 1,443 tokens for generation, with a constant denoising step independent of dataset statistics. Furthermore, following [Khan *et al.*, 2024], we redesign the data annotation pipeline for both DressCodeData [Korosteleva and Lee, 2021; He *et al.*, 2024] and GarmentCodeData, to provide both brief and detailed text descriptions for sewing patterns. To support the image modality as input, we employ commonly used garment sketches as the interactive interface between users and models. As a benefit of our modeling paradigm, we also support sewing pattern completion using user-provided incomplete patterns as input for controllable generation.

To sum up, our contributions to the community are as follows:

1. We present a new generative model, ***GarmentDiffusion***, pushing the limits of diffusion-based modeling paradigm for multimodal sewing pattern generation.
2. We propose an efficient edge encoding scheme that significantly reduces the token sequence length of sewing patterns, achieving a substantial speedup compared with the autoregressive approach, i.e., DressCode.
3. We validate the effectiveness of our model on SewFactory, DressCodeData, as well as the largest and most challenging GarmentCodeData, and establish a strong baseline with comprehensive and quantitative evaluation metrics.
4. We provide new multimodal data annotation pipelines that can generate both brief and detailed text descriptions, as well as garment sketches for sewing patterns, enabling multimodal sewing pattern generation.

2 Related Work

2.1 Garment Sewing Pattern Generation

Existing research on garment generation can mainly be divided into 3D-based and sewing pattern-based approaches. The 3D-based methods generates garment models through Gaussian splatting guidance [Li *et al.*, 2024], unsigned distance function regression [Moon *et al.*, 2022; Zheng *et al.*, 2024], neural volumetric rendering [Chen *et al.*, 2024b], or latent representation learning [Su *et al.*, 2023; Shen *et al.*, 2020; Srivastava *et al.*, 2025; Shao *et al.*, 2024]. However, these garment models often have topological imperfections that make them unsuitable for manufacturing.

To generate production-ready sewing patterns, early methods included scanned model flattening [Bang *et al.*, 2021] and iterative panel parameter optimization [Wang *et al.*, 2018]. With the release of large sewing pattern datasets [Korosteleva and Lee, 2021; Korosteleva *et al.*, 2024], there has been a shift toward data-driven approaches. SPnet [Lim *et al.*, 2024] predicts sewing patterns from generated T-pose images, while Neural Sewing Machine [Chen *et al.*, 2022] utilizes principal component analysis to create sewing pattern masks. To reduce reliance on templates, Korosteleva and Lee [2022] predict edges and stitches directly from 3D point clouds. Liu et al. [2023] and Chen et al. [2024a] use hierarchical transformers to recover sewing patterns from images. He et al. [2024] generate garment sewing patterns with textures, guided by natural language descriptions. At the same time as our work, Design2GarmentCode [Zhou *et al.*, 2024] proposes a DSL-oriented multimodal agent and leverage the garment programs to generate sewing patterns. ChatGarment [Bian *et al.*, 2024] fine-tunes a VLM to generate garment specification files. AIpparel [Nakayama *et al.*, 2024] builds on top of LLaVA-1.5 [Liu *et al.*, 2024a] and proposes to train a multimodal pattern generation model using both discrete and continuous training objectives. SewingLDM [Liu *et al.*, 2024b] applies a latent diffusion model with two-stage training to generate patterns, incorporating various handcrafted losses. Our model adopts a single-stage training with MSE loss, achieving improvements in modality diversity, performance, and efficiency.

2.2 Conditional Diffusion Models

A large portion of current research on diffusion models originates from [Ho *et al.*, 2020; Rombach *et al.*, 2022], with a for-

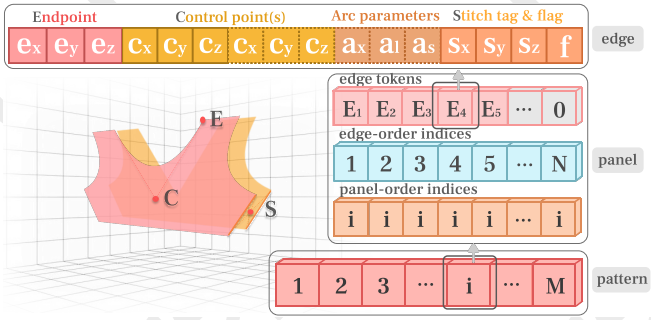


Figure 2: **Token representations for the edge, panel and pattern.** GarmentDiffusion utilizes an edge-oriented compact representation to encode the sewing pattern. After applying rotation and translation transformations to the 2D panels, the edge parameters are encoded along the embedding dimension. Each edge token is assigned an edge-order index and a panel-order index to indicate its global position within the sequence. The sequence is padded with zero tokens to ensure uniform length.

ward chain that perturbs data into noise and a reverse chain that converts noise back into data. Diffusion Transformers (DiT) [Peebles and Xie, 2023] explore replacing the U-Net backbone with a transformer that operates on latent patches, achieving better scalability. IP-Adapter [Ye *et al.*, 2023] introduces the decoupled cross-attention mechanism to achieve the image-prompt conditional generation.

Diffusion models have also proven successful in generating CAD models. For example, Xu et al. [2024] present a diffusion-based approach that unconditionally outputs boundary representation of CAD models. Wang et al. [2024] utilize a vector-quantized diffusion model to generate command sequences from design concepts represented by text or sketches. The success of diffusion models in synthesizing structured CAD models suggests that similarly structured sewing patterns could also be generated through a reverse diffusion process.

3 Method

3.1 Sewing Pattern Representation

Edge representation. A 3D pattern consists of variable number of panels, with each panel being a closed shape made up of multiple edges. Since the edges are connected end-to-end, we only need a start point $\mathbf{e}_j \in \mathbb{R}^3$ and control points $\mathbf{c}_j \in \mathbb{R}^3$ of Bézier curves to represent the geometry of j^{th} edge E_j, where $\mathbf{e}_j := (e_x, e_y, e_z)_j$ and $\mathbf{c}_j := (c_x, c_y, c_z)_j$ are the 3D point coordinates. The number of control points depends on whether the Bézier curve is quadratic or cubic. We represent the circular arc using another three parameters $\mathbf{a}_j \in \mathbb{R}^3$, where $\mathbf{a}_j := (a_x, a_l, a_s)_j$ represents the radius, major or minor arc and sweeping orientation, respectively. Note that if E_j is a linear curve, $\mathbf{c}_j$ is identical to $\mathbf{e}_j$. Furthermore, we use a per-edge stitch tag $\mathbf{s}_j \in \mathbb{R}^3$ and a binary stitch flag $f_j \in \{0, 1\}$ to encode the stitch information of edges. $\mathbf{s}_j := (s_x, s_y, s_z)_j$ is calculated as the averaged 3D midpoint between matched edge pairs. All coordinates are calculated after performing 3D rotation and translation for each panel. E_j is thus represented as $\mathbf{e}_j \oplus \mathbf{c}_j \oplus \mathbf{a}_j \oplus \mathbf{s}_j \oplus f_j$ with the appropriate zero padding, where $\oplus$ represents the concatenation along the embedding dimension. Note that the number of parameters of E_j, denoted as $|E_j|$, could be variable length depending on the dataset.

Pattern representation. Suppose that a dataset contains at most M panels for all patterns, and each panel contains at most N edges. We pad the pattern $\mathbf{P} \in \mathbb{R}^{|m \times n| \times |E_j|}$ that has m panels ($m \leq M$) and n edges per panel ($n \leq N$) to the uniform $M \times N$ sequence length, which is denoted as $\mathbf{P}' \in \mathbb{R}^{|M \times N| \times |E_j|}$. That is, all panels $\{P_i\}_{i>m}^{M}$ are set to $\mathbf{0}$, and all edges $\{E_j\}_{j>n}^{N}$ are set to $\mathbf{0}$ as well, where i and j denote the index of panel and edge, respectively. The hierarchical pattern representation is illustrated in Figure 2. This edge-oriented pattern representation largely shortens the token sequence length compared to the coordinate-oriented representation in DressCode. Different from the "token-by-token" causal generation, all these tokens can be processed by subsequent diffusion transformers **in parallel**.

3.2 Pattern Generation with GarmentDiffusion

Our model follows the design of DiT [Peebles and Xie, 2023] architecture. It accepts multimodal inputs to control the generation of sewing patterns. During the training phase, all edges are converted into token representations, followed by random panel shuffling and noise corruption. The model is trained to predict the noises added to the edge tokens. In the generation phase, the edge tokens are initialized as random Gaussian noises, and are iteratively denoised using the predicted noise. The entire framework is illustrated in Figure 3.

Pattern preprocessing. Suppose we have $\mathbf{P}' \in \mathbb{R}^{|M \times N| \times |E_j|}$, which consists of $\{P_i\}_{i=1}^{M}$ with $\{P_i\}_{i>m}^{M}$ being panel-level padding. Each P_i consists of $\{E_j\}_{j=1}^{N}$ with $\{E_j\}_{j>n}^{N}$ being edge-level padding. We shift and scale each dimension of E_j using respective parameter statistics to ensure that the value range is between -1 and 1. To support the pattern completion, we randomly shuffle $\{P_i\}_{i=1}^{m}$ to break the predefined panel order, while keeping the order of edges $\{E_j\}_{j=1}^{n}$ within a panel unchanged. We always place the padding tokens after the shuffled edge tokens.

Token embeddings. To distinguish the edge tokens among different panels, we construct a panel-level look-up embedding table $\mathbf{Emb}_P \in \mathbb{R}^{M \times C}$, where C is the embedding dimension of the model. We also construct an edge-level look-up embedding table $\mathbf{Emb}_E \in \mathbb{R}^{N \times C}$ to encode the sequential edge order within a panel. To represent each time step t, we use the conventional sine and cosine positional encoding to construct the time embedding $\mathbf{t} \in \mathbb{R}^C$. We employ a DDPM noise scheduler to obtain the noise-corrupted edge token, that is $\texttt{ddpm\_scheduler.add\_noise}(E_j, \boldsymbol{\epsilon}, t) \rightarrow \tilde{E}_j$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Finally, we compute the edge token embedding as:

$$\mathbf{x}_j = \varphi(\tilde{E}_j) + \mathbf{Emb}_P(i) + \mathbf{Emb}_E(j) + \mathcal{T}(\mathbf{t}), \quad (1)$$

where $\varphi(\cdot)$ and $\mathcal{T}(\cdot)$ are the projections, each comprising two linear layers with a non-linear activation function, to match the model's dimension.

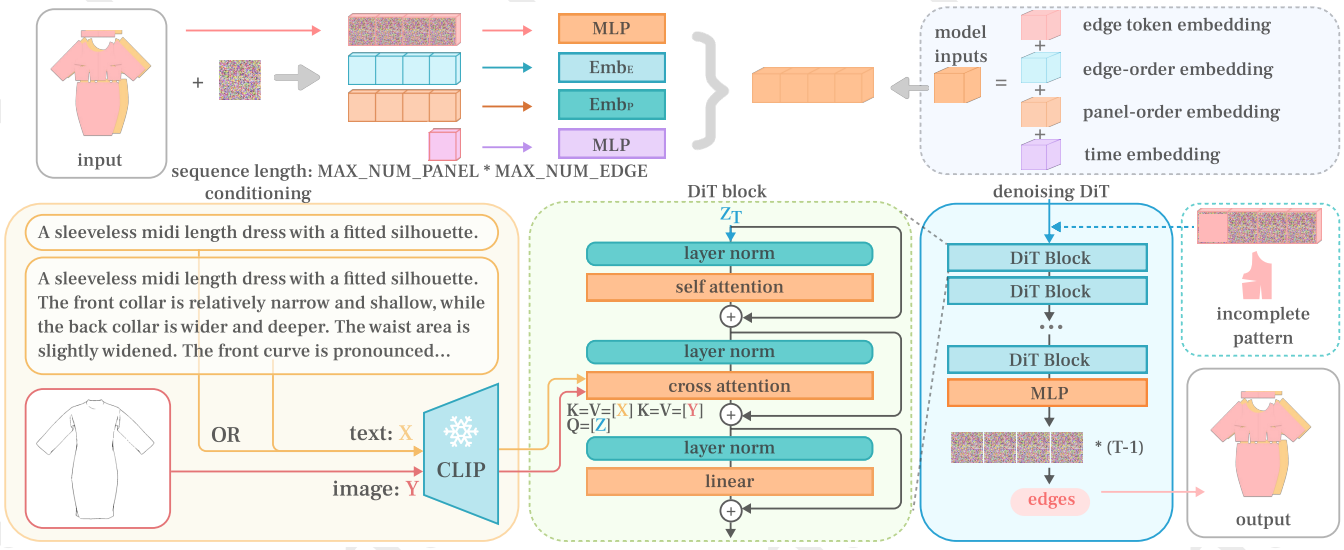
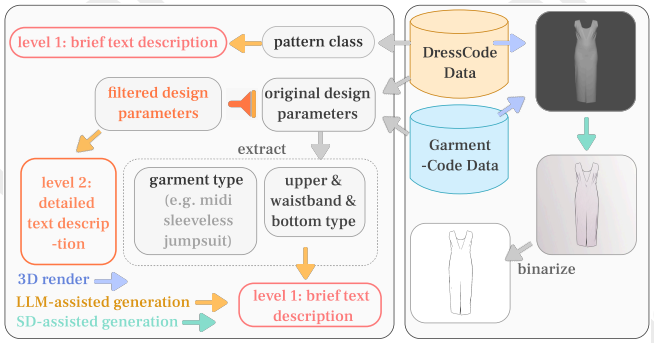


Figure 3: **The framework of GarmentDiffusion.** GarmentDiffusion accepts multi-level text descriptions, a garment sketch or an incomplete sewing pattern as input conditions, and generates a sewing pattern through the denoising of random Gaussian noise. The text and image features are extracted using a frozen CLIP and injected via decoupled cross-attention layers in each DiT block. The incomplete pattern replaces the initial subsequence of the random noise sequence for controllable generation. The final output is a 3D-placed sewing pattern that is consistent with the multimodal conditions.



SYSTEM_PROMPT
You are an experienced pattern maker. You must extract useful garment design information from a design parameters JSON file. Note that 'f_' means front, 'b_' means back.
USER_PROMPT
Your response must have these two parts:
1. "abstract": Use one sentence (not dict) to describe the garment type and component type in the "meta" section of the JSON. Do not describe features.
2. "part_params": Use a whole paragraph (not dict) to analyze the parameters for each part of the garment, the description is based on their values' relative positions within the specified range. Please adjust the adjectives accordingly to match the parameter names. Do not use numerical descriptions. Do not describe the type of the parts.
Use a whole paragraph as the value of "abstract", "part_types" or "part_params", not dict.
output format:
{"abstract":"<garment and component type>",
"part_params":"< parameter analysis>"}
The JSON object: json
Please extract information from it using the above methods. Strictly output json string, no opening statement or any explanation.

level 1: brief text description
A mini skirt pattern with a straight waistband and circular skirt components.

level 2: detailed text description
The waistband is narrow, with a moderate waistline. The skirt features a slightly shortened length, a standard rise, and moderate suns. The asymmetrical front is relatively short, while the cut is deep and moderately wide, positioned near the center.

Figure 4: **Our multimodal data annotation pipelines.** The up-left pipeline illustrates the generation of text descriptions at both brief and detailed levels, and the up-right pipeline represents the generation of garment sketches. The system and user prompts for LLM and text descriptions examples are shown at the bottom.

Conditional training. To achieve both text-to-pattern and image-to-pattern generation with fine-grained control, we inject the conditions using cross-attention layers rather than adaLN-Zero layers [Peebles and Xie, 2023]. Specifically, we follow the practice [Ye *et al.*, 2023] to employ decoupled cross-attention layers that use separate key and value projection matrices to process text and image features, while using shared query projection matrices among different modalities to process edge features. The multimodal cross-attention operation is defined as:

$$\mathbf{Z}' = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}_T^\top}{\sqrt{C}}\right)\mathbf{V}_T + \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}_I^\top}{\sqrt{C}}\right)\mathbf{V}_I, \tag{2}$$

where $\mathbf{Z}'$ is the fused multimodal latent features; $\mathbf{Q}$ is the edge features from $\mathbf{Z}$ after query projection; $\mathbf{K}_T$, $\mathbf{V}_T$, $\mathbf{K}_I$, $\mathbf{V}_I$ are the text and image features after key and value projection. The text and image features are extracted using CLIP's text and image encoders [Radford *et al.*, 2021]. We use both class and patch embeddings before the last projection layers of CLIP for the conditional training. The image features are projected into the same dimension as the text features by a two-layer MLP before fed into the diffusion transformer. All the parameters of the diffusion transformer are trainable. The training objective of our model is a simple $L2$ loss, which minimizes the mean-squared error between the sampled Gaussian noise and the predicted noise. That is:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{x},\mathbf{c}_T,\mathbf{c}_I,\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I}),t}\left[||\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}_T, \mathbf{c}_I, t)||_2^2\right], \tag{3}$$

where $\mathbf{x}_t$ is the edge token embeddings at time step t; $\mathbf{c}_T$ and $\mathbf{c}_I$ are the text and image conditions, respectively.

Method & dataset	$PanelL2 \downarrow$	$\#Panel \uparrow$	$\#Edge \uparrow$	$RotL2 \downarrow$	$TransL2 \downarrow$	$Precision \uparrow$	$Recall \uparrow$	$F1 \uparrow$
SewingGPT & original	1.02e1	0.754	0.887	7.51e-3	1.25e0	0.833	0.833	0.825
SewingGPT & brief	8.89e0	0.936	0.951	8.51e-3	9.85e-1	0.933	0.933	0.933
SewingGPT & detailed	8.35e0	0.979	0.946	8.34e-3	8.93e-1	0.973	0.974	0.973
Ours & original	8.24e0	0.794	0.969	**9.21e-5**	9.42e-1	0.856	0.857	0.849
Ours & brief	7.48e0	0.955	0.999	1.99e-4	7.79e-1	0.956	0.955	0.955
Ours & detailed	**6.53e0**	**0.989**	**0.999**	2.79e-4	**7.30e-1**	**0.989**	**0.989**	**0.989**

Table 1: **Quantitative evaluation results on the DressCodeData (test set).** The metrics are explained in Section 4.3. The $L2$ metrics are measured in centimeters (except $RotL2$). The first three rows show SewingGPT's evaluation results using its original captions (generated by GPT-4V) and ours (brief and detailed descriptions). The last three rows show GarmentDiffusion's evaluation results with the same text conditions. Our model outperforms SewingGPT by a large margin on different levels of captions and captions generated by different pipelines.

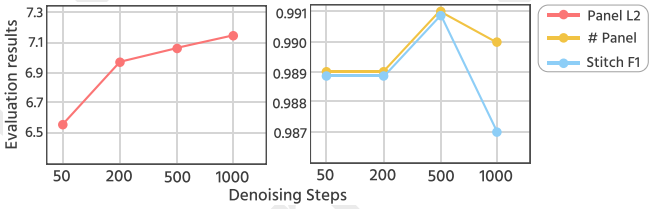


Figure 5: **Comparison of model performance with different denoising steps on the DressCodeData.** Increasing the number of denoising steps does not lead to improved model performance.

4 Experiments

4.1 Datasets

We use SewFactory [Liu *et al.*, 2023], DressCodeData [Korosteleva and Lee, 2021; He *et al.*, 2024] and GarmentCodeData (**V2**) [Korosteleva *et al.*, 2024] for training and evaluation. For SewFactory, we employ off-the-shelf rendered garments superimposed on diverse human poses as image prompts (without text prompts). For DressCodeData and GarmentCodeData, we designed multimodal data annotation pipelines (depicted in Figure 4) to generate both text and image prompts for sewing patterns. SewFactory consists of 13,707 sewing patterns, featuring a maximum of 14 panels and 12 edges per panel. DressCodeData contains 19,683 patterns, each with up to 10 panels and 10 edges per panel. GarmentCodeData offers 115,195 patterns, with a maximum of 37 panels and 39 edges per panel. Since the official splits of SewFactory are not available, we use our own version that 90% of randomly selected data points are used for training, with the remaining 10% evenly divided for validation and testing. For DressCodeData and GarmentCodeData (V2), we adhere strictly to the official splits provided by the authors for training, validation, and testing. Note that SewingGPT cannot be trained on the entire GarmentCodeData due to its long context length. To address this problem, we created a subset of GarmentCodeData by filtering out those patterns with `#edges/panel > 12` and `#panels/pattern > 10`.

4.2 Multimodal Data Synthesis

Text-prompt generation. To enhance our model's comprehension of multi-level design concepts from users, we design two-level text descriptions for the sewing patterns: a concise category-level summary with basic design features, and detailed component-level descriptions. Leveraging LLMs' in-context understanding, our annotation pipeline first filters irrelevant information from garment specification files, then prompts Llama-3.1-8B-Instruct [Grattafiori *et al.*, 2024] using a unified prompt to generate the multi-level descriptions automatically, as shown in Figure 4.

Image-prompt generation. We select the garment sketches as the interactive interface between users and our model. To generate the garment sketches that closely emulate the hand-drawn style of professional garment designers, we initially render the 3D garment models (in `.obj` and `.ply` formats) of DressCodeData and GramentCodeData into 2D garment images using Blender's APIs [BlenderAuthors, 2024]. Subsequently, we utilize MistoLine [Zhang *et al.*, 2023] and Anything-XL fine-tuned from SD-XL [Podell *et al.*, 2023] to extract the garment sketches, followed by a binarization operation.

4.3 Evaluation Metrics

We adopt the same evaluation metrics as [Liu *et al.*, 2023] and [Korosteleva and Lee, 2022] to assess the fidelity of the generated sewing patterns. Specifically, **Panel L2** denotes the $L2$ distance of the coordinates of 2D panels (converted from 3D coordinates) between predictions and ground truths, with the centroids of panels shifted to the origin. **#Panel** and **#Edge** represent the accuracy of correctly predicted patterns within all patterns, based on the number of panels in each pattern and the number of edges in each panel, respectively. **Rot L2** and **Trans L2** represent the $L2$ distances of x, y, z rotation Euler angles and universal x, y, z translations of panels between predictions and ground truths. **Precision**, **Recall**, and **F1 Score** are used to measure the false positives and false negatives of paired stitches with respect to all edge relations.

4.4 Implemention Details

Architecture. We adopt OpenAI ViT-H/14 (336×) [Radford *et al.*, 2021] as our text and image encoders. Since the embedding dimensions of the text and image features are 1,024 and 768, we project the image features into 768-dimensional vectors to match the text feature dimension. The main body of our model consists of 12 DiT blocks. Each block consists of a self-attention layer, a multimodal cross-attention layer and a feed-forward layer, all utilizing pre-layer normalization [Ba *et al.*, 2016; Xiong *et al.*, 2020]. The num-

Input	$PL2 \downarrow$	$\#Panel \uparrow$	$\#Edge \uparrow$	$RotL2 \downarrow$	$TrsL2 \downarrow$	$F1 \uparrow$
SewingGPT & brief	1.52e1	0.708	0.686	1.52e-2	2.71e0	0.529
SewingGPT & detailed	1.34e1	0.762	0.733	1.52e-2	2.03e0	**0.589**
Ours & brief	1.37e1	0.738	0.706	2.26e-3	1.90e0	0.463
Ours & detailed	**1.19e1**	**0.815**	**0.786**	**1.50e-3**	**1.74e0**	0.553

Table 2: **Quantitative evaluation results on the subset of GarmentCodeData.** Due to the high memory requirements for training caused by excessively long sequences in SewingGPT, we filtered the GarmentCodeData to retain only sewing patterns with no more than 10 panels and a maximum of 12 edges per panel (consistent with SewingGPT) for training.

Input	$PL2 \downarrow$	$\#Panel \uparrow$	$\#Edge \uparrow$	$RotL2 \downarrow$	$TrsL2 \downarrow$	$F1 \uparrow$
SewFormer†	3.76e0	0.859	0.956	2.01e-2	6.10e-1	**0.946**
Ours	**3.73e0**	**0.883**	**0.978**	**3.51e-3**	**6.03e-1**	0.942

Table 3: **Quantitative evaluation results on the SewFactory (test set).** Note that the official training split of SewFactory is not provided by authors. Therefore, SewFormer† is retrained by us using its official codebase with our split and evaluated on the same test set.

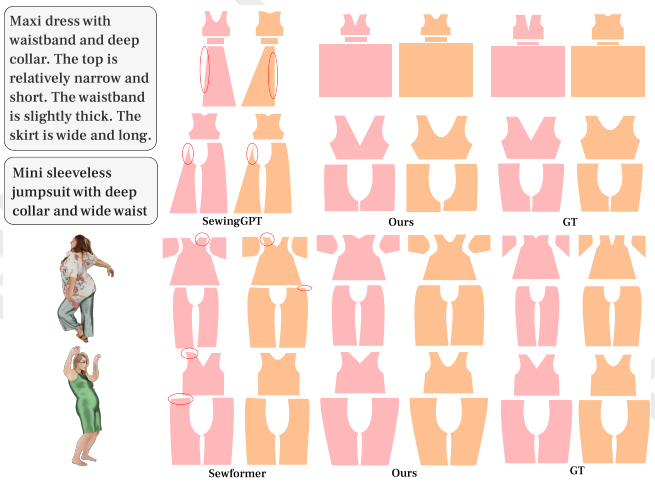


Figure 6: **Visualization of patterns generated with SewingGPT, SewFormer and ours.** Major errors of the baseline approaches are highlighted with red circles.

ber of heads for each attention layer is set to 8. The embedding dimension of the DiT blocks is 768, while the feed-forward layers have an embedding dimension of $1,024$.

Look-up embedding tables. The panel-level and edge-level embedding tables contain M and N learnable positional embeddings, where M is the maximum number of panels and N is the maximum number of edges per panel in each dataset. For SewFactory, $M = 14$, $N = 12$. For DressCodeData, $M = N = 10$. For GarmentCodeData, $M = 37$, $N = 39$.

Training details. We adopt a DDPM noise scheduler for diffusion training, with a maximum of $1,000$ denoising steps and a linear beta scheduler (beta_start $= 1\times10^{-4}$, beta_end $= 2 \times 10^{-2}$). We use the AdamW optimizer [Loshchilov and Hutter, 2019] with betas $= (0.95, 0.999)$, a constant learning rate of 1×10^{-4} and the weight decay of 1×10^{-2}. The training epoch is set to $1,000$ with an early-stop criterion. We evaluate the model at denoising steps of 50, 200, 500, and 1000 every 10 epochs. Based on the results shown in Figure 5, we select 50 denoising steps for inference. The multimodal training is performed in a round-robin fashion, following the order of image prompts, text prompts and image-and-text prompts. Our model is distributedly trained across 8 A10 GPUs (24GB) with the Hugging Face Accelerate library [Gugger *et al.*, 2022].

4.5 Comparison with State-of-the-Art Methods

Compared with SewingGPT. Our evaluation metrics assess the accuracy of geometric structures, panel placement in 3D space, and stitching relations. Table 1 shows the evaluation results for SewingGPT and GarmentDiffusion, using text prompts from two different annotation pipelines. The first three rows show that the text descriptions generated by GPT-4V with DressCode's pipeline result in worse performance than ours. This is expected, as we prompt the Llama3.1-8B-Instruct (text-only) [MetaAI, 2024] with precise design specifications, which are used to generate sewing patterns through programs. Moreover, our method outperforms SewingGPT when trained with the same captions, highlighting the advantage of our approach.

We also evaluated SewingGPT on the subset of GarmentCodeData. As shown in Table 2, our method outperforms SewingGPT in panel and edge number accuracies, geometrical shapes, and 3D placement of panels. However, it lags in stitching edge prediction, likely due to the lack of stitching information in the prompt.

As shown in Figure 7, our model also supports pattern completion with incomplete patterns provided by users. It achieves strong control even though the text descriptions are not precise.

Compared with SewFormer. We also trained SewFormer and GarmentDiffusion using our training split as described in Section 4.1, while maintaining the same evaluation protocols and test set to ensure fair comparison. Table 3 demonstrates that our generative model achieves comparable performance to SewFormer in terms of the geometrical shapes and 3D panel placement, while it surpasses SewFormer in terms of panel and edge accuracies. These results confirm the effectiveness of our method, even when using multi-pose rendered images as prompts.

4.6 Ablation Study

Inputs of different modalities. To assess the impact of training on model performance under different combinations of input modalities, we trained three models using text, image, and multimodal prompts on the GarmentCodeData, respectively. As shown in Table 4, under the multimodal training setting, evaluation metrics are gradually improved when fine-grained conditions are incorporated for generation. Specifically, the combination of detailed text descriptions and sketches achieve the best performance. This conclusion is also supported by the middle section of the table. Compared with the model trained using text-only prompts, the model trained with image-only prompts exhibits slightly better per-

Train Modality	Text		Image	$PanelL2 \downarrow$	$\#Panel \uparrow$	$\#Edge \uparrow$	$RotL2 \downarrow$	$TransL2 \downarrow$	$Precision \uparrow$	$Recall \uparrow$	$F1 \uparrow$
	brief	detailed	sketch								
MM (text&image)	✓			1.31e1	0.388	0.600	2.26e-3	1.69e0	0.380	0.364	0.364
MM (text&image)		✓		1.12e1	0.464	0.701	2.07e-3	1.53e0	0.472	0.457	0.460
MM (text&image)			✓	1.08e1	0.537	0.713	**1.70e-3**	1.38e0	0.430	0.431	0.425
MM (text&image)	✓		✓	7.48e0	0.616	0.771	1.90e-3	1.00e0	0.516	0.506	0.509
MM (text&image)		✓	✓	**6.68e0**	**0.670**	**0.819**	1.85e-3	**9.26e-1**	**0.564**	**0.561**	**0.560**
Text-only	✓			1.52e1	0.287	0.470	3.13e-3	2.00e0	0.270	0.239	0.244
Text-only		✓		1.08e1	0.486	0.707	2.60e-3	1.61e0	0.474	0.473	0.469
Image-only			✓	1.05e1	0.528	0.723	1.91e-3	1.31e0	0.443	0.436	0.435

Table 4: **Quantitative evaluation results of GarmentDiffusion on the GarmentCodeData (test set).** The first five rows are trained using both text and image prompts and evaluated with different modality combinations. The two rows in the middle are trained with text prompts only and evaluated with brief or detailed text descriptions. The last row is trained using image prompts. We use the whole GarmentCodeData for training.

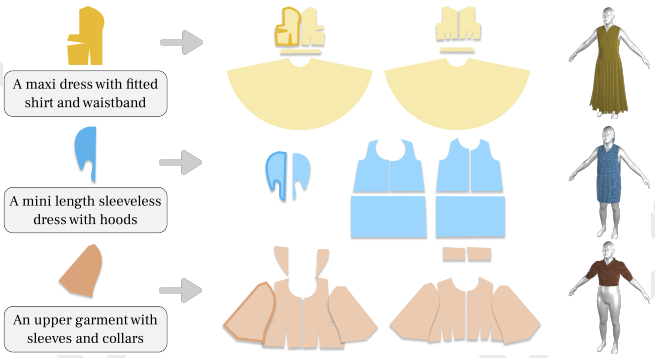


Figure 7: **Visualization of pattern completion.** Given a ground-truth panel and a text prompt, our model is capable of generating a consistent and complete sewing pattern.

formance. This is reasonable, as images may convey more information than texts.

Condition injection method. In addition to employing cross-attention for conditional training, we also experimented with adaLN-Zero conditioning proposed in DiT [2023]. We trained multimodal models (text & image) with both conditioning methods on the augmented DressCodeData. For cross-attention, we evaluate the model under text, image, and text-and-image conditions, while for adaLN-Zero, we used text and image conditions with equal probability. As shown in Table 5, although adaLN-Zero exhibits advantages when evaluated with brief text descriptions, it becomes less effective with detailed text descriptions and image inputs due to insufficient conditional information extraction. Cross-attention achieves the best overall performance when both detailed text and image inputs are provided.

4.7 Limitations and Future Works

While current annotations provide detailed sewing pattern descriptions, they still lack stitching information on edge and panel connectivity, which can compromise garment simulation. The annotation engine thus remains improvable. Additionally, current methods offer limited control via numerical parameters (e.g., panel/edge count) or human body measurements. From an efficiency standpoint, reducing denoising steps is also desirable. Future work will target these challenges by enhancing controllability and generation efficiency.

Condition Scheme	Text		Image	$PanelL2 \downarrow$	$\#Panel \uparrow$	$F1 \uparrow$
	brief	detailed	sketch			
Cross-attention	✓			7.42e0	0.944	0.949
		✓		6.64e0	0.994	0.995
			✓	2.47e0	0.978	0.985
	✓		✓	2.41e0	0.994	0.994
		✓	✓	**2.39e0**	**0.994**	**0.995**
AdaLN-Zero	✓			7.49e0	0.958	0.949
		✓		7.54e0	0.970	0.959
			✓	3.25e0	0.968	0.979

Table 5: **Quantitative evaluation results for different condition injection methods.** The first five rows present the evaluation results using the cross-attention method, while the last three rows correspond to the conditioning using the adaLN-Zero method. Both methods utilizes text and image prompts for training. We report three metrics to save the space in the table.

5 Conclusion

In conclusion, we introduced GarmentDiffusion, a much under-explored research direction for sewing pattern generation. Our model incorporates the design of diffusion transformers with an efficient edge encoding scheme. The architecture and training of our model are simple yet efficient, enabling the end-to-end generation of centimeter-precise and vectorized 3D sewing patterns. Our experimental results demonstrate the effectiveness of our method, which bridges the gap between creative garment design and manufacturing through scalable, precise, and efficient generative modeling. This work lays the foundation for advancing AI-driven fashion technology, seamlessly connecting digital design with practical garment production.

References

[Ba *et al.*, 2016] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[Bang *et al.*, 2021] Seungbae Bang, Maria Korosteleva, and Sung-Hee Lee. Estimating garment patterns from static scan data. In *Computer Graphics Forum*, volume 40, pages 273–287. Wiley Online Library, 2021.

[Bian *et al.*, 2024] Siyuan Bian, Chenghao Xu, Yuliang Xiu, Artur Grigorev, Zhen Liu, Cewu Lu, Michael J. Black, and Yao Feng. ChatGarment: Garment estimation, generation and editing via large language models, 2024.

[BlenderAuthors, 2024] BlenderAuthors. Blender python api. https://docs.blender.org/api/current/info_overview.html, 2024.

[Carion *et al.*, 2020] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[Chen *et al.*, 2022] Xipeng Chen, Guangrun Wang, Dizhong Zhu, Xiaodan Liang, Philip Torr, and Liang Lin. Structure-preserving 3D garment modeling with neural sewing machines. *Advances in Neural Information Processing Systems*, 35:15147–15159, 2022.

[Chen *et al.*, 2024a] Cheng-Hsiu Chen, Jheng-Wei Su, Min-Chun Hu, Chih-Yuan Yao, and Hung-Kuo Chu. Panelformer: Sewing pattern reconstruction from 2D garment images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 454–463, January 2024.

[Chen *et al.*, 2024b] Yizheng Chen, Rengan Xie, Sen Yang, Linchen Dai, Hongchun Sun, Yuchi Huo, and Rong Li. Single-view 3D garment reconstruction using neural volumetric rendering. *IEEE Access*, 2024.

[Grattafiori *et al.*, 2024] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughana, and et al. The Llama 3 herd of models, 2024.

[Gugger *et al.*, 2022] Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

[He *et al.*, 2024] Kai He, Kaixin Yao, Qixuan Zhang, Jingyi Yu, Lingjie Liu, and Lan Xu. DressCode: Autoregressively sewing and generating garments from text guidance. *ACM Transactions on Graphics (TOG)*, 43(4):1–13, 2024.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[Khan *et al.*, 2024] Mohammad Sadil Khan, Sankalp Sinha, Talha Uddin Sheikh, Didier Stricker, Sk Aziz Ali, and Muhammad Zeshan Afzal. Text2CAD: Generating sequential CAD models from beginner-to-expert level text prompts. *arXiv preprint arXiv:2409.17106*, 2024.

[Korosteleva and Lee, 2021] Maria Korosteleva and Sung-Hee Lee. Generating datasets of 3D garments with sewing patterns, 2021.

[Korosteleva and Lee, 2022] Maria Korosteleva and Sung-Hee Lee. NeuralTailor: Reconstructing sewing pattern structures from 3D point clouds of garments. *ACM Transactions on Graphics (TOG)*, 41(4):1–16, 2022.

[Korosteleva and Sorkine-Hornung, 2023] Maria Korosteleva and Olga Sorkine-Hornung. GarmentCode: Programming parametric sewing patterns. *ACM Transactions on Graphics*, 42(6):1–15, December 2023.

[Korosteleva *et al.*, 2024] Maria Korosteleva, Timur Levent Kesdogan, Fabian Kemper, Stephan Wenninger, Jasmin Koller, Yuhan Zhang, Mario Botsch, and Olga Sorkine-Hornung. GarmentCodeData: A dataset of 3D made-to-measure garments with sewing patterns, 2024.

[Li *et al.*, 2024] Boqian Li, Xuan Li, Ying Jiang, Tianyi Xie, Feng Gao, Huamin Wang, Yin Yang, and Chenfanfu Jiang. GarmentDreamer: 3DGS guided garment synthesis with diverse geometry and texture details, 2024.

[Lim *et al.*, 2024] Seungchan Lim, Sumin Kim, and Sung-Hee Lee. SPnet: Estimating garment sewing patterns from a single image of a posed user. In *Eurographics (Short Papers)*, 2024.

[Liu *et al.*, 2023] Lijuan Liu, Xiangyu Xu, Zhijie Lin, Jiabin Liang, and Shuicheng Yan. Towards garment sewing pattern reconstruction from a single image. *ACM Transactions on Graphics*, 42(6):1–15, December 2023.

[Liu *et al.*, 2024a] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024.

[Liu *et al.*, 2024b] Shengqi Liu, Yuhao Cheng, Zhuo Chen, Xingyu Ren, Wenhan Zhu, Lincheng Li, Mengxiao Bi, Xiaokang Yang, and Yichao Yan. Multimodal latent diffusion model for complex sewing pattern generation, 2024.

[Loshchilov and Hutter, 2019] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[MetaAI, 2024] MetaAI. Llama-3.1-8b instruct model. Hugging Face Model Hub, 2024.

[Moon *et al.*, 2022] Gyeongsik Moon, Hyeongjin Nam, Takaaki Shiratori, and Kyoung Mu Lee. 3D clothed human reconstruction in the wild. In *European conference on computer vision*, pages 184–200. Springer, 2022.

[Nakayama *et al.*, 2024] Kiyohiro Nakayama, Jan Ackermann, Timur Levent Kesdogan, Yang Zheng, Maria Korosteleva, Olga Sorkine-Hornung, Leonidas J. Guibas, Guandao Yang, and Gordon Wetzstein. AIpparel: A large multimodal generative model for digital garments, 2024.

[Peebles and Xie, 2023] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision*, pages 4195–4205, 2023.

[Podell *et al.*, 2023] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis, 2023.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[Rombach *et al.*, 2022] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[Shao *et al.*, 2024] Liangjing Shao, Benshuang Chen, Ziqun Zhang, and Xinrong Chen. 3D clothed human model reconstruction based on single-view in-the-wild image data. *IEEE Sensors Journal*, 2024.

[Shen *et al.*, 2020] Yu Shen, Junbang Liang, and Ming C. Lin. GAN-based garment generation using sewing pattern images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[Srivastava *et al.*, 2025] Astitva Srivastava, Pranav Manu, Amit Raj, Varun Jampani, and Avinash Sharma. WordRobe: Text-guided generation of textured 3D garments. In *European Conference on Computer Vision*, pages 458–475. Springer, 2025.

[Su *et al.*, 2023] Zhaoqi Su, Tao Yu, Yangang Wang, and Yebin Liu. DeepCloth: Neural garment representation for shape and style editing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1581–1593, February 2023.

[Wang *et al.*, 2018] Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popović, and Niloy J. Mitra. Learning a shared shape space for multimodal garment design. *ACM Trans. Graph.*, 37(6), December 2018.

[Wang *et al.*, 2024] Hanxiao Wang, Mingyang Zhao, Yiqun Wang, Weize Quan, and Dong-Ming Yan. VQ-CAD: Computer-aided design model generation with vector quantized diffusion. *Computer Aided Geometric Design*, 111:102327, 2024.

[Xiong *et al.*, 2020] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020.

[Xu *et al.*, 2024] Xiang Xu, Joseph Lambourne, Pradeep Jayaraman, Zhengqing Wang, Karl Willis, and Yasutaka Furukawa. BrepGen: A B-rep generative diffusion model with structured latent geometry. *ACM Transactions on Graphics (TOG)*, 43(4):1–14, 2024.

[Ye *et al.*, 2023] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

[Zhang *et al.*, 2023] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *International Conference on Computer Vision*, pages 3836–3847, 2023.

[Zheng *et al.*, 2024] Jiali Zheng, Rolandos Alexandros Potamias, and Stefanos Zafeiriou. Design2Cloth: 3D cloth generation from 2D masks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1748–1758, June 2024.

[Zhou *et al.*, 2024] Feng Zhou, Ruiyang Liu, Chen Liu, Gaofeng He, Yong-Lu Li, Xiaogang Jin, and Huamin Wang. Design2GarmentCode: Turning design concepts to tangible garments through program synthesis, 2024.